applied sciences

MDPI

Article

A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection

Ibomoiye Domor Mienye *,† **and Yanxia Sun** †

Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2006, South Africa
* Correspondence: ibomoiyem@uj.ac.za
† These authors contributed equally to this work.

Abstract: With the rapid developments in electronic commerce and digital payment technologies, credit card transactions have increased significantly. Machine learning (ML) has been vital in analyzing customer data to detect and prevent fraud. However, the presence of redundant and irrelevant features in most real-world credit card data degrades the performance of ML classifiers. This study proposes a hybrid feature-selection technique consisting of filter and wrapper feature-selection steps to ensure that only the most relevant features are used for machine learning. The proposed method uses the information gain (IG) technique to rank the features, and the top-ranked features are fed to a genetic algorithm (GA) wrapper, which uses the extreme learning machine (ELM) as the learning algorithm. Meanwhile, the proposed GA wrapper is optimized for imbalanced classification using the geometric mean (G-mean) as the fitness function instead of the conventional accuracy metric. The proposed approach achieved a sensitivity and specificity of 0.997 and 0.994, respectively, outperforming other baseline techniques and methods in the recent literature.

Keywords: credit card; feature selection; fraud detection; genetic algorithm; machine learning

Citation: Mienye, I.D.; Sun, Y. A Machine Learning Method with Hybrid Feature Selection for Improved Credit Card Fraud Detection. *Appl. Sci.* **2023**, *13*, 7254. https://doi.org/10.3390/app13127254

Academic Editor: Luigi Portinale

Received: 18 March 2023
Revised: 9 June 2023
Accepted: 13 June 2023
Published: 18 June 2023

1. Introduction

Over the years, electronic payments (e-payments) have been the most common payment option due to technological advancements and the development of several electronic funding methods [1]. E-payment systems are essential to the present competitive financial sector and are mostly performed using credit cards [2]. The introduction of credit cards has resulted in convenient and seamless e-payments. A recent study stated that in the second quarter of 2021, Mastercard and Visa issued 1131 million and 1156 million cards, respectively [3]. However, the rise of credit card usage globally has increased the fraud rate, affecting consumers and merchants [4]. For instance, a report stated that financial losses due to credit and debit cards are among the leading causes of losses in the financial sector [3]. Therefore, developing efficient credit card fraud-detection systems is necessary to reduce such losses.

Machine learning algorithms have been widely employed to detect credit card fraud [5–7]. Meanwhile, there have been enormous datasets with very high dimensions due to the advent of big data and the Internet of Things (IoT) [8,9]. Furthermore, some features in these datasets might be redundant or less significant to the response variable. Using such features for machine learning could increase the complexity of the model and lead to overfitting [10]. Therefore, to handle the high dimensionality issue, an approach containing dimensionality reduction, such as feature selection, is necessary to obtain valuable insights and make accurate predictions [11].

Feature-selection techniques aim to identify the most important attributes needed to develop a well-performing machine learning model [12,13], ensuring improved classification performance and reduced computational complexity by removing irrelevant and

redundant features. Feature selection techniques are usually grouped into three methodological groups: filters, wrappers, and embedded methods [10,14]. The internal workings and configuration of the various feature-selection methods make them suitable for different applications. Filter methods employ attribute ranking to determine the most informative features. Features that attain scores above a given threshold are selected, and those below the threshold are discarded. After identifying the most important features, they can be fed as input to the learning algorithm. Filter methods vary from wrapper and embedded methods as they are not dependent on a classifier and are, therefore, independent of the classifier's bias [15].

However, wrapper methods use an ML classifier's performance as the evaluation metric in selecting the most relevant feature set. Wrapper methods usually lead to better classification performance than filter techniques because the feature-selection procedure is optimized for the chosen classification algorithm [16,17]. Generally, wrapper methods employ a search strategy to identify the candidate subsets. The classifier's performance on the various feature subsets is measured, and the subset that leads to the highest performance is selected as the most informative subset. Examples of wrapper-based feature selection techniques include the Boruta algorithm, forward selection, backward elimination, and the genetic algorithm. Embedded methods select the features that enhance the model's performance during training. The feature selection is incorporated into the learning procedure [13]. Unlike wrapper methods, this type of feature selection aims to reduce the time used in training different subsets. Embedded methods include random forest, decision tree, gradient boosting, elastic net, and LASSO [10].

Meanwhile, the GA wrapper is an effective method for feature selection, with applications in diverse domains, including natural language processing (NLP) [18], fraud detection [19], sentiment analysis [20], and medical diagnosis [21]. This study proposes a hybrid feature-selection approach, combining the IG-based filter and GA-based wrapper techniques. The main contributions and objectives of the work include the following:

- Using the information gain technique for initial feature selection to rank the features in the credit card dataset, only the top-ranked features are fed into the GA wrapper to reduce the search space and enhance the classification performance.
- Secondly, the GA wrapper is employed to select the best feature subset that results in optimal classification performance, and the ELM is employed as the learning algorithm in the GA wrapper.
- Additionally, this study employs the G-mean as the fitness function in the GA wrapper instead of the conventional accuracy evaluation criterion, ensuring the recognition rate of the minority samples is considered and improved.

The rationale behind this approach is that the initial IG-based feature selection and ELM's ability to produce promising performance while converging faster than traditional neural networks could reduce the computational complexity of the GA and improve the classification performance. The ELM is chosen as the learning algorithm in the GA wrapper because it converges far more rapidly and achieves higher generalization performance than conventional neural networks. At the same time, its learning process is thousands of times quicker than neural networks trained via backpropagation [22]. Furthermore, for convenience, the proposed hybrid approach is called IG-GAW. It would be compared with the conventional ELM classifier, an ELM classifier with IG-based feature selection (IG-ELM), the GA wrapper (GAW), and well-performing methods in related literature.

The rest of this paper is structured as follows: Section 2 presents related works, focusing on feature-selection methods in the literature. Section 3 discusses the dataset and algorithms used in this study. The proposed credit card fraud-prediction approach is introduced in Section 4. Section 5 presents the results, while Section 6 concludes the paper and provides appropriate future research directions.

2. Related Works

Recently, ML algorithms have been widely applied for credit card fraud detection [23–25]. Researchers have used both traditional ML and deep learning (DL) algorithms to predict credit card fraud efficiently. For example, Alarfaj et al. [26] conducted a study using ML and DL techniques for detecting credit card fraud, while Van Belle et al. [27] employed inductive graph representation learning, Esenogho et al. [28] used a neural network ensemble, and Zhang et al. [29] employed an ensemble classifier based on isolation forest and adaptive boosting.

Some problems encountered when dealing with credit card datasets include high dimensionality and imbalance class [30,31], making it difficult for ML classifiers to learn and make accurate predictions. In addition, high dimensional data often make the learning process complex and computationally expensive, resulting in models with poor generalization ability [32]. Therefore, feature selection is essential in such datasets to reduce the computational burden and enhance the model's generalization ability. For example, Chaquet-Ulldemolins et al. [33] recorded an increase in the classification performance of ML classifiers after introducing feature selection. Generally, feature-selection methods are useful in applications where the number of features affects the classifier's performance.

The wrapper feature-selection methods have been widely applied in numerous applications [34,35]. They compute the importance of each feature based on its usefulness when training the ML model. The primary components of a wrapper method are the learning classifier and search strategy. The wrapper technique exists as a wrapper around the learning classifier and uses the same classifier to select the most relevant features. Therefore, a robust learning classifier could enhance the wrapper-based feature selection. Furthermore, the search strategy employed in the wrapper could affect the feature selection, and using the right search strategy for a given application is crucial in obtaining good performance.

Evolutionary search techniques such as genetic algorithms can avoid becoming stuck in local optima. Unlike deterministic algorithms, they can identify reduced feature sets that can effectively represent the original feature set [36]. The GA-based wrapper can easily identify feature redundancy and correlations. In addition, selecting a suitable classifier is vital in developing robust GA wrapper models since the wrapper procedure is tied to the selected classifier's performance. However, there are specific issues to consider when selecting the classifier. Firstly, the classifier should be able to achieve good classification performance and have excellent generalization ability. Secondly, since the classifier would be used to train numerous subsets, it should have good training speed. Thirdly, the number of features in the various subsets might differ. Therefore, using the same model parameters might not be enough to obtain good performance in all the subsets [37]. Hence, it would be preferred to use a classifier that automatically updates the model parameters for every feature subset to achieve good performance.

Other recent methods for credit card fraud detection include a signal processing framework [38], signal processing on graphs [39], and a deep learning ensemble [40]. In addition, in the literature, several learning algorithms (such as decision tree [41], naïve Bayes [42], SVM [43], and random forest [44]) have been used as the classifier in the GA wrapper. However, these classifiers are not able to consider the issues mentioned above. Therefore, a hybrid wrapper approach that considers all the above-mentioned issues is proposed. The proposed approach employs the IG-based filter feature selection to rank the attributes, and only the top-ranked features would be used as input into the GA wrapper. Meanwhile, the GA wrapper employs the ELM as the learning classifier. The ELM can achieve excellent classification performance and generalization ability with an extremely fast learning speed compared to conventional training methods. Furthermore, unlike traditional neural networks based on backpropagation algorithms, the ELM's training process is entirely automatic and does not require it to be tuned iteratively.

3. Materials and Methods

3.1. Credit Card Dataset

The European cardholders dataset [45] is used in this study. It is publicly available and comprises 284807 transactions made by European cardholders in September 2013. The dataset has been widely used in different credit card fraud-detection studies [1,6,46]. It contains 492 fraudulent transactions, and the rest are legitimate transactions, i.e., only 0.17% of the dataset belongs to the minority class, and 99.83% belongs to the majority class; hence, the dataset is highly skewed and it is challenging for conventional ML algorithms to learn from the dataset.

Due to privacy concerns, the features in the dataset were anonymized as $V1, V2, \ldots,$ and $V28$, except for the "Time" and "Amount" features. The "Time" attribute indicates the seconds elapsed between a transaction and the first transaction in the dataset, whereas the "Amount" indicates the value of the transaction. Meanwhile, the "Class" attribute is the response variable, representing legitimate and fraudulent transactions and having values 0 and 1, respectively.

3.2. Information Gain

The information gain technique, or mutual information, is one of the most used filter criteria. The IG criterion is modeled after the concept of entropy in information theory [47]. The entropy measures the impurity or uncertainty in a group of observations, while information gain computes the decrease in entropy before and after adding an attribute. An attribute with a high IG value is usually preferred to those with low IG values. Assuming X and Y are features in a dataset, the information gain of X, given Y is represented mathematically, is:

$$G(X|Y) = H(X) - H(X|Y) \tag{1}$$

In (1), $H(X)$ denotes the entropy of X and $H(X|Y)$ is the conditional entropy for X given Y [48]. Meanwhile, $H(X)$ and $H(X|Y)$ can be represented as:

$$H(X) = -\sum_{x \in X} P(x) log_2(x) \tag{2}$$

$$H(X|Y) = -\sum_{x \in X} P(x) \sum_{y \in Y} P(x|y) log_2(P(x|y)) \tag{3}$$

Therefore, given two features X and Z, a response variable Y is more correlated to X than Z if $IG(X|Y) > IG(Z|Y)$ [49]. Lastly, the information gain technique considers each feature separately, computes the IG value, and outputs its importance to the response variable.

3.3. Genetic Algorithm

The genetic algorithm, inspired by genetics in biological systems, can perform well in high-dimensional feature-selection problems because of its robustness, making it suitable for credit card fraud detection. Given a set of candidate features (called the population), the GA finds the optimal solution via a series of iterative genetic operations. It is superior to most traditional search methods in three main areas: firstly, the GA conducts a parallel search all over the population of solutions; instead of optimizing its parameters, the GA uses chromosomes, an encoded form of a possible solution, to achieve faster convergence. Lastly, the GA employs a fitness value to identify a potential solution. The flowchart of the GA is shown in Figure 1.

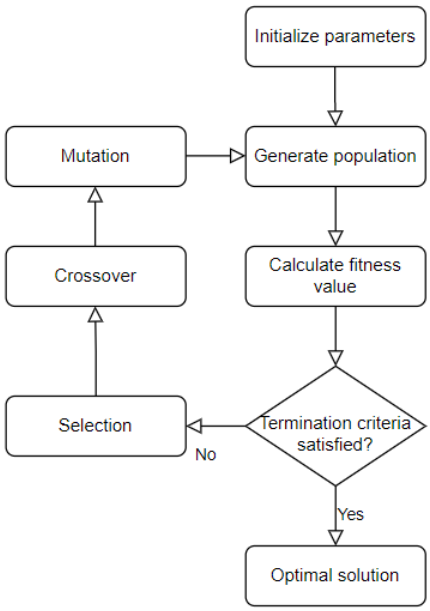


Figure 1. Genetic algorithm flowchart.

The genetic operators include crossover, mutation, and selection [50]. The crossover operator determines how the chromosomes are combined to obtain a new chromosome for the next generation [51]. In the GA, a population of candidate solutions (also known as individuals) to the optimization problem is evolved to obtain the optimal solution. The various candidate solutions have certain attributes called chromosomes that can be mutated; meanwhile, these solutions are represented in 0 and 1 binary strings, indicating whether the corresponding attribute has been selected or not. The mutation operator ensures some bits of the chromosomes are flipped randomly based on probability. The selection step involves choosing chromosomes based on their fitness score for further processing.

A common termination criterion is running the algorithm for a specified number of times [52]. Therefore, the algorithm ends after the specified number of iterations, outputting the optimal solution identified after going through all the generations.

3.4. Extreme Learning Machine

The ELM was developed by Huang et al. [50] to fix the slow learning speed of feed-forward neural networks. The authors attributed the slow learning speed to the use of gradient descent-based learning algorithms for training neural networks and how such algorithms iteratively tune the neural network parameters [22]. The ELM has excellent generalization ability with extremely fast learning.

Unlike traditional neural networks, the hidden layer parameters of the ELM are randomly generated without being iterative tuned [37], thereby reducing the learning procedure to just estimating the optimal output weights β. For a given dataset $(x_j, t_j)_{(j=1)}^{N}$, where N is the number of instances and the SLFN has L hidden nodes, the activations function $g(x)$ can be mathematically represented as:

$$\sum_{i=1}^{L} \beta_i g(w_i.x_j + b_i) = t_j \tag{4}$$

where t_j is the output of the network, $w_i = [w_{i1}, \ldots, w_{in}]^T$ represent the input weight, b_i denotes the bias of the $i - th$ hidden node, $\beta_i = [\beta_{i1}, \ldots, \beta_{im}]^T$ denotes the weight vector

linking the $i-th$ hidden node with the output nodes, and $w_i.x_j$ represents the inner product of w_i and x_j [53]. Equation (4) can be rewritten in the compact matrix form as

$$H\beta = T \tag{5}$$

where H represents the output matrix of the hidden layer and its mathematical formulation, presented by Huang et al. [22], is expressed as

$$\mathrm{H}(w_1,\ldots,w_N,b_1,\ldots,b_N,x_1,\ldots,x_N) = \begin{bmatrix} g(w_i.x_i+b_i) & \cdots & g(w_N.x_i+b_i) \\ \vdots & \ddots & \vdots \\ g(w_1.x_N+b_1) & \cdots & g(w_N.x_N+b_N) \end{bmatrix}_{N\times N} \tag{6}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{N\times m} \quad and \quad \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m} \tag{7}$$

4. Proposed Credit Card Fraud-Detection Approach

This study employs a hybrid feature-selection method, combining the IG-based filter and GA-based wrapper methods. Firstly, the IG technique ranks the attributes to identify the most significant attributes, and the threshold value is obtained by computing the standard deviation [1] of the IG values. Therefore, features with IG values greater or equal to the threshold are selected, while those below the threshold are discarded. The standard deviation has been widely employed in the literature to achieve excellent feature-selection thresholds [54–56]. Secondly, the top-ranked features are fed into the GAW, which uses the ELM as the learning algorithm.

The GA aims to identify the best feature subsets from a series of combinations known as generations [57]. Furthermore, after the ELM classifies the chromosomes, they are assigned a fitness value. Generally, the accuracy metric has been used as the standard fitness function [58]. However, it gives equal importance to samples in both majority and minority classes; hence, it is unsuitable for imbalanced classification problems [59].

Since the majority class samples outnumber the minority class, the fitness function will be biased toward the former. Furthermore, since there are more majority class samples, the accuracy metric will return high accuracy values, which could be misleading. Therefore, this study employs the G-mean metric obtained by the chromosomes as the fitness value to handle the imbalanced class. G-mean is a vital metric for imbalanced classification problems, and it considers the classifier's performance for the majority and minority classes. The G-mean can be represented mathematically as follows:

$$G\text{-}mean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{8}$$

where true positive (TP) indicates a fraudulent transaction that is correctly predicted by the model, false positive (FP) indicates a legitimate transaction that the model wrongly predicts as fraud, true negative (TN) indicates a legitimate transaction that is correctly identified, and false negative (FN) indicates a fraudulent transaction that is predicted as legitimate. The proposed approach is outlined in Algorithm 1.

Algorithm 1 Proposed IG-GAW

1. Compute the information gain of the features in the dataset
2. Rank features according to their importance: $F = (f_1 > f_2 > f_3, \ldots, f_n)$.
3. **GAW input:** Top-ranked features from Step 2 and class variable C, population size n, elitism rate e, number of iterations k.
4. Begin
5. Initialize population with n random solutions
6. Compute the fitness values for each random solution.
7. **for** $i = 1, \ldots, k$:
8. Select the best individuals with respect to e
9. Generate new offspring based on the GA operators
10. Examine the fitness values of the new individuals.
11. Discard least-fit population individuals in the population
12. **end for**
13. **GAW output:** Optimal solution S

The GAW approach starts with randomly initializing the GA's population, where every candidate feature subset is encoded as a chromosome. The next step involves training different ELM neural networks based on each chromosome, and the fitness value for each feature subset is computed. Thirdly, a new population is generated using genetic operators. The procedure continues until the stopping criterion is obtained, i.e., the maximum number of generations. This stopping criterion has been used extensively in the literature to obtain excellent GA performance [60–62]. Meanwhile, the rationale behind this approach is that the GA wrapper would select the best feature subset that would lead to enhanced prediction performance.

5. Results and Discussion

The proposed method's classification performance and other baseline classifiers are presented and discussed in the section. The machine learning models were implemented using scikit-learn [63], a widely used library for machine learning in Python. Meanwhile, the proposed method is compared with the following baseline classifiers: AdaBoost [64], logistic regression (LR) [65], random forest (RF) [66], SVM [67], and decision tree [68]. Furthermore, the stratified 10-fold cross-validation method is employed to measure the performance of the prediction models. The stratified k-fold technique ensures that the proportion of fraudulent and legitimate instances in the dataset is preserved in each fold, and it is usually more suitable for imbalance classification problems than the k-fold cross-validation method [69].

The following metrics are used to evaluate the performance of the models: sensitivity, specificity, the receiver operating characteristic (ROC) curve, and the area under the ROC curve (AUC). Sensitivity refers to the model's ability to predict a fraud transaction as fraud. Usually, a highly sensitive model is preferred in fraud detection, as it implies there are no or few false negative predictions [70,71]. Meanwhile, the specificity of a model is its ability to predict non-fraudulent transactions as legitimate. A model with high specificity means there are hardly any false positive predictions [72]. Sensitivity and specificity can be computed as follows:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{10}$$

The ROC curve is used to visualize the performance of binary classifiers. It plots the true positive rate against the false positive rate at various classification thresholds [73]. In contrast, the AUC summarizes the ROC curve, and its value ranges from 0 to 1. An AUC of 0 implies the model's predictions are all wrong, and a value of 1 implies all the model

predictions are correct. The AUC is a crucial metric in imbalance classification problems, such as fraud detection, as it indicates the classifier's ability to differentiate between the fraud and non-fraud classes.

Furthermore, in line with existing literature, this study estimates the best GA parameters by conducting several trials using different combinations [74–76]. The final GA parameters used in this study are outlined in Table 1.

Table 1. GA parameters.

Parameter	Value
Population size	50
Number of generations	100
Crossover rate	0.6
Mutation rate	0.01
Fitness function	G-mean
Stopping criteria	Max number of generations
Type of mutation	Uniform mutation
Type of crossover	Single point
Parent selection method	Tournament selection
Tournament size	2

5.1. Performance of the ELM Classifier with Filter, Wrapper, and Hybrid Feature Selection Methods

The ELM's performance without feature selection is compared with instances where the ELM classifier is coupled with the filter, wrapper, and hybrid feature-selection techniques. Firstly, the performance of the ELM without feature selection is recorded. Secondly, the credit card features and their IG values are ranked by the IG-based filter technique.

The standard deviation [54] of the IG values is calculated and used as the threshold value to select the most informative features. From Table 2, the standard deviation is 0.145. Hence, information gain values above 0.145 are chosen as the essential features employed for training the machine learning model. The features with IG values below 0.145 are removed. Therefore, the top 21 attributes are chosen by the filter approach as the most important features, while the following attributes are removed: $V8$, $V19$, $V24$, $V23$, $V26$, $V13$, $V25$, $V15$, and $V22$.

Table 2. Performance of the ELM classifier, filter, wrapper, and hybrid feature-selection methods.

Classifier	Sensitivity	Specificity	AUC	G-Mean
ELM	0.881	0.904	0.900	0.892
IG-ELM	0.936	0.960	0.940	0.947
GAW	0.949	0.962	0.950	0.955
IG-GAW	0.997	0.994	0.990	0.994

Thirdly, the GA wrapper coupled with the ELM classifier (GAW) is trained, and its performance is recorded. Lastly, the top 21 features selected by the IG technique are used as input to the GAW. The performance from the four scenarios is tabulated in Table 2, i.e., ELM without feature selection, filter-based IG-ELM, wrapper-based GAW, and the hybrid IG-GAW method. In addition, Figure 2 shows the ROC curves and AUC of the various models. Meanwhile, the complete feature set and the features selected by the various feature-selection methods are tabulated in Table 3.

Table 2 and Figure 2 show that the proposed hybrid IG-GAW obtained the highest sensitivity, specificity, and AUC values of 0.997, 0.994, and 0.990, respectively, outperforming the ELM, the IG-ELM, and IG-GAW.

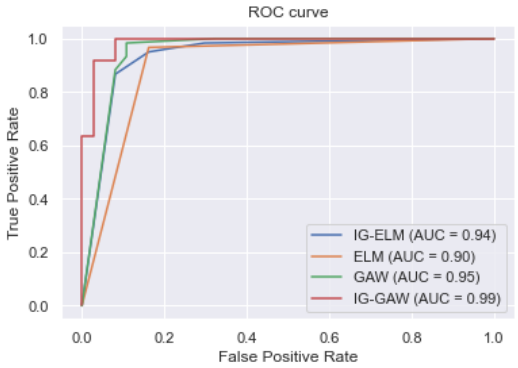


Figure 2. ROC curve of the ELM classifier, filter, wrapper, and hybrid feature-selection methods.

Table 3. Feature sets from the European cardholders dataset.

Feature-Selection Method	Features
Complete feature set	$V1$, $V2$, $V3$, $V4$, $V5$, $V6$, $V7$, $V8$, $V9$, $V10$, $V11$, $V12$, $V13$, $V14$, $V15$, $V16$, $V17$, $V18$, $V19$, $V20$, $V21$, $V22$, $V23$, $V24$, $V25$, $V26$, $V27$, $V28$, Time, Amount, Class
IG	$V1$, $V2$, $V3$, $V4$, $V5$, $V6$, $V7$, $V9$, $V10$, $V11$, $V12$, $V14$, $V16$, $V17$, $V18$, $V20$, $V21$, $V27$, $V28$, Time, Amount
GAW	$V1$, $V2$, $V5$, $V6$, $V7$, $V9$, $V10$, $V11$, $V12$, $V16$, $V17$, $V18$, $V20$, $V21$, $V27$, $V28$, Time, Amount
IG-GAW	$V1$, $V2$, $V5$, $V6$, $V7$, $V9$, $V11$, $V12$, $V16$, $V17$, $V18$, $V20$, $V21$, $V27$, $V28$

5.2. Performance Comparison with Baseline Classifiers and Recent Literature

In this section, the proposed hybrid approach is benchmarked with other ML classifiers and methods in the literature. The baseline classifiers were trained with the complete feature set. The performance of the classifiers is tabulated in Table 4 and visualized in Figure 3. The models obtained relatively high specificity compared to the sensitivity, which implies the model correctly predicted more non-fraud transactions (majority class) than fraud transactions (minority class).

Meanwhile, in credit card fraud detection, like every imbalance classification task, it is more important to predict the minority class samples correctly. However, the proposed method achieved excellent sensitivity and specificity, indicating its robustness in predicting the minority and majority class samples. This enhanced performance could be attributed to using the G-mean as the fitness function rather than the widely used accuracy criterion, ensuring the model's detection rate on the minority class is enhanced.

Table 4. Performance comparison with other baseline classifiers.

Classifier	Sensitivity	Specificity	AUC	G-Mean
AdaBoost	0.889	0.918	0.900	0.903
LR	0.752	0.916	0.810	0.829
RF	0.869	0.940	0.890	0.904
SVM	0.585	0.827	0.660	0.695
DT	0.590	0.801	0.690	0.688
Proposed IG-GAW	0.997	0.994	0.990	0.994

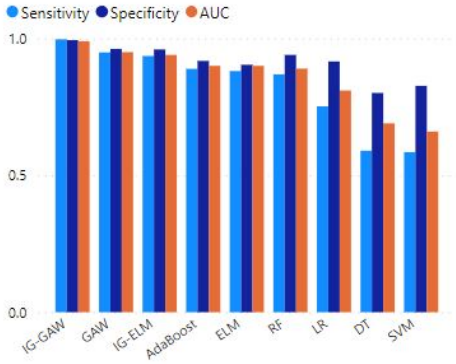


Figure 3. Comparative analysis of the various methods using the credit card dataset.

Furthermore, Table 5 shows the performance of some state-of-the-art methods in the literature. The methods include a weighted extreme learning machine (weighted ELM) [77], a deep neural network (DNN)-based classifier [78], a cost-sensitive neural network ensemble (CS-NNE) [79], a random forest-based genetic algorithm wrapper method (GA-RF) [19], a method that sequentially combines C4.5 and the naïve Bayes model (DT-NB) [80], a model developed using the random forest algorithm with the SMOTE technique (RF-SMOTE) [81], a stochastic ensemble model [82], an XGBoost-SMOTE model [83], a support vector machine (SVM)-based GA wrapper [84], an ensemble model optimized using the particle swarm optimization (PSO) technique [85], a metaheuristic based feature selection algorithm based on rock hyrax swarm optimization (RHSO) [86], and a deep residual network with shuffled shepherd optimization (DRN-SSPO) [87].

Table 5. Performance comparison with other well-performing methods in the literature.

Reference	Algorithm	Sensitivity	Specificity	AUC
Zhu et al. [77]	Weighted ELM	0.982	-	0.978
Alkhatib et al. [78]	DNN	0.955	-	0.990
Yotsawat et al. [79]	CS-NNE	-	0.936	0.980
Ileberi et al. [19]	GA-RF	72.56	-	0.950
Kalid et al. [80]	DT-NB	0.872	1.000	-
Mrozek et al. [81]	Random forest-SMOTE	0.829	-	0.910
Carta et al. [82]	Stochastic ensemble	0.915	-	0.876
Xie et al. [83]	XGBoost-SMOTE	0.988	-	0.970
Saheed et al. [84]	GA-SVM	0.963	0.963	-
Verma et al. [85]	PSO-based Ensemble model	0.97	-	-
Padhi et al. [86]	RHSO	0.951	-	-
Ganji et al. [87]	DRN-SSPO	0.912	0.902	-
This paper	Proposed IG-GAW	0.997	0.994	0.990

Table 5 shows that the proposed IG-GAW demonstrated excellent results compared to the state-of-the-art methods, indicating the proposed method's robustness. Furthermore, the proposed method is applied to other credit card datasets to show how the method performs in different scenarios. The datasets include the German credit card dataset [88] and the Taiwan credit card dataset [89]. The German dataset has 20 attributes and contains 1000 samples, of which 700 are classified as good and 300 as bad customers, i.e., 70% of the dataset belongs to the majority class, while 30% belongs to the minority class; hence, the dataset is imbalanced. The features in the German dataset and those selected by the various feature selection techniques are shown in Table 6, while Table 7 and Figure 4 show the performance of the various models.

Table 6. Feature sets from the German dataset.

Feature-Selection Method	Features
Complete feature set	Status of existing checking account, duration in month, credit history, purpose, credit amount, savings account, present employment since, installment rate as a percentage of disposable income, personal status and sex, other debtors, present residence since, property, age, other installment plans, housing, number of existing credits at this bank, job, number of dependents, telephone, foreign worker
IG	Status of existing checking account, duration in month, credit history, purpose, credit amount, savings account, present employment since, installment rate as a percentage of disposable income, personal status and sex, other debtors, property, age, other installment plans, housing, number of dependents, foreign worker
GAW	Status of existing checking account, duration in month, credit history, purpose, credit amount, savings account, present employment since, property, age, other installment plans, housing, number of dependents, foreign worker
IG-GAW	Credit amount, status of existing checking account, duration in months, age, credit history, purpose, property, present employment since, and housing

Table 7. Performance comparison using the German dataset.

Classifier	Sensitivity	Specificity	AUC	G-Mean
AdaBoost	0.785	0.892	0.810	0.837
LR	0.688	0.813	0.700	0.748
RF	0.796	0.904	0.830	0.850
SVM	0.649	0.792	0.650	0.716
DT	0.630	0.787	0.640	0.704
ELM	0.704	0.830	0.710	0.763
IG-ELM	0.796	0.903	0.810	0.847
GAW	0.820	0.925	0.860	0.871
Proposed IG-GAW	0.904	0.946	0.910	0.925

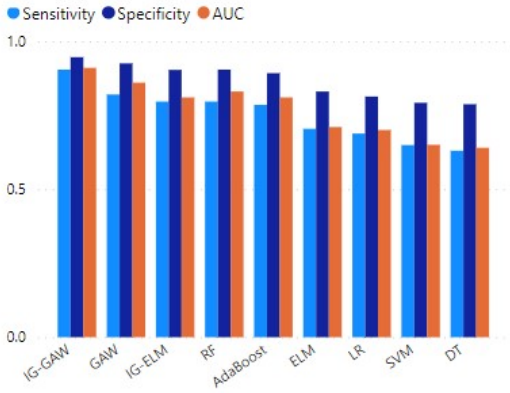


Figure 4. Comparative analysis of the various methods using the German dataset.

Meanwhile, the Taiwan dataset has 25 attributes and 30,000 samples, of which 23,364 are classified as good and 6636 as bad customers, i.e., 77.88% of the dataset belongs to the majority class and 22.12% belongs to the minority class. Therefore, the Taiwan dataset is also imbalanced. The features in the datasets and the selected feature sets are shown in Table 8. Meanwhile, Table 9 and Figure 5 show the performance of the various methods.

Table 8. Feature sets from the Taiwan dataset.

Feature Selection Method	Features
Complete feature set	ID, LIMIT_BAL, SEX, EDUCATION, MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6
IG	SEX, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6
GAW	PAY_0, PAY_2, PAY_4, PAY_5, BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT4, PAY_AMT5, PAY_AMT6
IG-GAW	BILL_AMT1, BILL_AMT2, BILL_AMT3, BILL_AMT4, BILL_AMT5, BILL_AMT6, PAY_AMT1, PAY_AMT2, PAY_AMT3, PAY_AMT6, PAY_AMT4, PAY_AMT5, PAY_0, and PAY_2

Table 9. Performance comparison using the Taiwan dataset.

Classifier	Sensitivity	Specificity	AUC	G-Mean
AdaBoost	0.870	0.890	0.870	0.880
LR	0.625	0.837	0.640	0.723
RF	0.829	0.914	0.840	0.870
SVM	0.626	0.819	0.650	0.716
DT	0.574	0.773	0.610	0.666
ELM	0.710	0.885	0.730	0.793
IG-ELM	0.874	0.911	0.890	0.892
GAW	0.899	0.920	0.900	0.909
Proposed IG-GAW	0.945	0.961	0.940	0.952

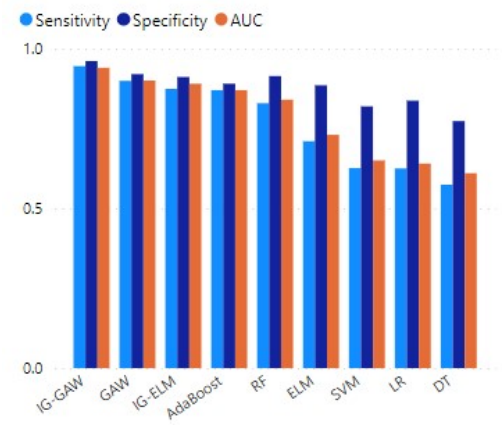


Figure 5. Comparative analysis of the various methods using the Taiwan dataset.

Using the German dataset, the proposed IG-GAW achieved a sensitivity of 0.904, specificity of 0.945, and AUC of 0.910, as shown in Table 7. Meanwhile, the experimental results for the Taiwan dataset are shown in Table 9, and it shows that the proposed method obtained a sensitivity of 0.945, specificity of 0.961, and AUC of 0.940.

5.3. Discussions

Credit card fraud is a huge burden for financial institutions. The advances in e-commerce and digital payment platforms have made credit card fraud more common. This study aimed to utilize machine learning to detect credit card fraud effectively. Meanwhile, since most credit card datasets contain irrelevant attributes that degrade the performance of machine learning algorithms, this study proposed a robust hybrid feature-selection approach comprising filter (information gain technique) and wrapper (genetic algorithm) feature-selection steps, ensuring only the most significant attributes are used for machine learning. This study uses the well-known European Credit Card dataset. The proposed IG-GAW, which uses the ELM as the learning algorithm in the GA wrapper, obtained scores for sensitivity, specificity, AUC, and G-mean of 0.997, 0.994, 0.990, and 0.994, respectively. Additionally, two popular credit risk datasets (German and Taiwan credit datasets) were used to further validate the proposed method's improved performance.

The proposed IG-GAW outperformed the selected classifiers when trained with the German and Taiwan datasets. The results also showed that the proposed hybrid approach, IG-GAW, achieved superior performance compared to the filter and wrapper methods, i.e., IG-ELM and GAW. Therefore, integrating filter and wrapper techniques in a hybrid setting is a robust approach to detecting credit card fraud. Lastly, the results also showed that introducing feature selection enhanced the ELM's performance, indicating the importance of effective feature selection.

6. Conclusions

Detecting fraudulent credit card transactions is challenging, and researchers have developed different methods to handle this problem. This study proposed a hybrid approach to enhance the detection rate. The hybrid approach takes advantage of the strength of different feature-selection and ML methods, including information gain, genetic algorithms, and extreme learning machines. The IG technique was employed for initial feature selection, and the top-ranked features served as input to the GA wrapper. Meanwhile, the ELM was used as the learning algorithm in the GA wrapper. The proposed approach outperformed other baseline classifiers and methods in recent literature. Furthermore, the proposed method was applied to two more credit card datasets to validate its performance, and it achieved excellent performance in both datasets, demonstrating its robustness. Therefore, it can be concluded that the proposed hybrid approach is an effective credit card fraud-detection method. Future research work would employ more combinations of evolutionary algorithms and ML-based feature selection methods to enhance new aspects of credit card fraud detection. In addition, future research would explore the potential of obtaining more recent datasets to train ML models.

Author Contributions: Conceptualization, I.D.M. and Y.S.; methodology, I.D.M. and Y.S.; software, I.D.M.; validation, I.D.M. and Y.S.; formal analysis, I.D.M. and Y.S.; investigation, I.D.M. and Y.S.; resources, Y.S.; data curation, I.D.M. and Y.S.; writing—original draft preparation, I.D.M.; writing—review and editing, I.D.M. and Y.S.; visualization, I.D.M. and Y.S.; supervision, Y.S.; funding acquisition, Y.s. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the South African National Research Foundation under Grant 120106 and Grant 132797 and in part by the South African National Research Foundation Incentive under Grant 132159.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Femila Roseline, J.; Naidu, G.; Samuthira Pandi, V.; Alamelu alias Rajasree, S.; Mageswari, N. Autonomous credit card fraud detection using machine learning approach. *Comput. Electr. Eng.* **2022**, *102*, 108132. [CrossRef]
2. Alharbi, A.; Alshammari, M.; Okon, O.D.; Alabrah, A.; Rauf, H.T.; Alyami, H.; Meraj, T. A Novel text2IMG Mechanism of Credit Card Fraud Detection: A Deep Learning Approach. *Electronics* **2022**, *11*, 756. [CrossRef]
3. Bin Sulaiman, R.; Schetinin, V.; Sant, P. Review of Machine Learning Approach on Credit Card Fraud Detection. *Hum.-Centric Intell. Syst.* **2022**, *2*, 55–68. [CrossRef]
4. Wang, D.; Chen, B.; Chen, J. Credit card fraud detection strategies with consumer incentives. *Omega* **2019**, *88*, 179–195. [CrossRef]
5. Nandi, A.K.; Randhawa, K.K.; Chua, H.S.; Seera, M.; Lim, C.P. Credit card fraud detection using a hierarchical behavior-knowledge space model. *PLoS ONE* **2022**, *17*, e0260579. [CrossRef]
6. Ileberi, E.; Sun, Y.; Wang, Z. Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. *IEEE Access* **2021**, *9*, 165286–165294. [CrossRef]
7. Rtayli, N.; Enneya, N. Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. *J. Inf. Secur. Appl.* **2020**, *55*, 102596. [CrossRef]
8. Oo, M.C.M.; Thein, T. An efficient predictive analytics system for high dimensional big data. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 1521–1532. [CrossRef]
9. Huebner, J.; Fleisch, E.; Ilic, A. Assisting mental accounting using smartphones: Increasing the salience of credit card transactions helps consumer reduce their spending. *Comput. Hum. Behav.* **2020**, *113*, 106504. [CrossRef]
10. Pudjihartono, N.; Fadason, T.; Kempa-Liehr, A.W.; O'Sullivan, J.M. A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. *Front. Bioinform.* **2022**, *2*, 927312. [CrossRef]
11. de-la-Bandera, I.; Palacios, D.; Mendoza, J.; Barco, R. Feature Extraction for Dimensionality Reduction in Cellular Networks Performance Analysis. *Sensors* **2020**, *20*, 6944. [CrossRef]
12. Bouaguel, W. A New Approach for Wrapper Feature Selection Using Genetic Algorithm for Big Data. In *Intelligent and Evolutionary Systems*; Springer : Cham, Switzerland, 2016; pp. 75–83. [CrossRef]
13. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [CrossRef]
14. Bashir, S.; Khattak, I.U.; Khan, A.; Khan, F.H.; Gani, A.; Shiraz, M. A Novel Feature Selection Method for Classification of Medical Data Using Filters, Wrappers, and Embedded Approaches. *Complexity* **2022**, *2022*, e8190814. [CrossRef]
15. Kumar, A.; Bhatia, M.P.S.; Sangwan, S.R. Rumour detection using deep learning and filter-wrapper feature selection in benchmark twitter dataset. *Multimed. Tools Appl.* **2022**, *81*, 34615–34632. [CrossRef]
16. Wang, F.; Lu, X.; Chang, X.; Cao, X.; Yan, S.; Li, K.; Duić; N; Shafie-khah, M.; Catalão, J.P. Household profile identification for behavioral demand response: A semi-supervised learning approach using smart meter data. *Energy* **2022**, *238*, 121728. [CrossRef]
17. Wang, Z.; Gao, S.; Zhou, M.; Sato, S.; Cheng, J.; Wang, J. Information-Theory-based Nondominated Sorting Ant Colony Optimization for Multiobjective Feature Selection in Classification. *IEEE Trans. Cybern.* **2022**, 1–14. [CrossRef]
18. Rasool, A.; Tao, R.; Kamyab, M.; Hayat, S. GAWA–A Feature Selection Method for Hybrid Sentiment Classification. *IEEE Access* **2020**, *8*, 191850–191861. [CrossRef]
19. Ileberi, E.; Sun, Y.; Wang, Z. A machine learning based credit card fraud detection using the GA algorithm for feature selection. *J. Big Data* **2022**, *9*, 24. [CrossRef]
20. Al-Ahmad, B.; Al-Zoubi, A.M.; Abu Khurma, R.; Aljarah, I. An Evolutionary Fake News Detection Method for COVID-19 Pandemic Information. *Symmetry* **2021**, *13*, 1091. [CrossRef]
21. Soumaya, Z.; Drissi Taoufiq, B.; Benayad, N.; Yunus, K.; Abdelkrim, A. The detection of Parkinson disease using the genetic algorithm and SVM classifier. *Appl. Acoust.* **2021**, *171*, 107528. [CrossRef]
22. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990. [CrossRef]
23. Han, S.; Zhu, K.; Zhou, M.; Cai, X. Competition-Driven Multimodal Multiobjective Optimization and Its Application to Feature Selection for Credit Card Fraud Detection. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 7845–7857. [CrossRef]
24. Malik, E.F.; Khaw, K.W.; Belaton, B.; Wong, W.P.; Chew, X. Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. *Mathematics* **2022**, *10*, 1480. [CrossRef]
25. Zioviris, G.; Kolomvatsos, K.; Stamoulis, G. Credit card fraud detection using a deep learning multistage model. *J. Supercomput.* **2022**, *78*, 14571–14596. [CrossRef]
26. Alarfaj, F.K.; Malik, I.; Khan, H.U.; Almusallam, N.; Ramzan, M.; Ahmed, M. Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms. *IEEE Access* **2022**, *10*, 39700–39715. [CrossRef]
27. Van Belle, R.; Van Damme, C.; Tytgat, H.; De Weerdt, J. Inductive Graph Representation Learning for fraud detection. *Expert Syst. Appl.* **2022**, *193*, 116463. [CrossRef]
28. Esenogho, E.; Mienye, I.D.; Swart, T.G.; Aruleba, K.; Obaido, G. A Neural Network Ensemble with Feature Engineering for Improved Credit Card Fraud Detection. *IEEE Access* **2022**, *10*, 16400–16407. [CrossRef]
29. Zhang, Y.-F.; Lu, H.-L.; Lin, H.-F.; Qiao, X.-C.; Zheng, H. The Optimized Anomaly Detection Models Based on an Approach of Dealing with Imbalanced Dataset for Credit Card Fraud Detection. *Mob. Inf. Syst.* **2022**, *2022*, e8027903. [CrossRef]

30. Ala'raj, M.; Abbod, M.F.; Majdalawieh, M.; L. Jum'a. A deep learning model for behavioural credit scoring in banks. *Neural Comput. Appl.* **2022**, *34*, 5839–5866. [CrossRef]
31. Zhang, X.; Yu, L.; Yin, H.; Lai, K.K. Integrating data augmentation and hybrid feature selection for small sample credit risk assessment with high dimensionality. *Comput. Oper. Res.* **2022**, *146*, 105937. [CrossRef]
32. Yang, Y.; Fan, C.; Chen, L.; Xiong, H. IPMOD: An efficient outlier detection model for high-dimensional medical data streams. *Expert Syst. Appl.* **2022**, *191*, 116212. [CrossRef]
33. Chaquet-Ulldemolins, J.; Gimeno-Blanes, F.-J.; Moral-Rubio, S.; Muñoz-Romero, S.; Rojo Álvarez, J.-L. On the Black-Box Challenge for Fraud Detection Using Machine Learning (I): Linear Models and Informative Feature Selection. *Appl. Sci.* **2022**, *12*, 3328. [CrossRef]
34. Al-Yaseen, W.L.; Idrees, A.K.; Almasoudy, F.H. Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. *Pattern Recognit.* **2022**, *132*, 108912. [CrossRef]
35. Beheshti, Z. BMPA-TVSinV: A Binary Marine Predators Algorithm using time-varying sine and V-shaped transfer functions for wrapper-based feature selection. *Knowl.-Based Syst.* **2022**, *252*, 109446. [CrossRef]
36. Prashanth, S.K.; Shitharth, S.; Praveen Kumar, B.; Subedha, V.; Sangeetha, K. Optimal Feature Selection Based on Evolutionary Algorithm for Intrusion Detection. *SN Comput. Sci.* **2022**, *3*, 439. [CrossRef]
37. Xue, X.; Yao, M.; Wu, Z. A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm. *Knowl. Inf. Syst.* **2018**, *57*, 389–412. [CrossRef]
38. Salazar, A.; Safont, G.; Rodriguez, A.; Vergara, L. Combination of multiple detectors for credit card fraud detection. In Proceedings of the 2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Limassol, Cyprus, 12–14 December 2016; pp. 138–143. [CrossRef]
39. Vergara, L.; Salazar, A.; Belda, J.; Safont, G.; Moral, S.; Iglesias, S. Signal processing on graphs for improving automatic credit card fraud detection. In Proceedings of the 2017 International Carnahan Conference on Security Technology (ICCST), Madrid, Spain, 23–26 October 2017; pp. 1–6. [CrossRef]
40. Mienye, I.D.; Sun, Y. A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection. *IEEE Access* **2023**, *11*, 30628–30638. [CrossRef]
41. Gkikas, D.C.; Theodoridis, P.K.; Beligiannis, G.N. Enhanced Marketing Decision Making for Consumer Behaviour Classification Using Binary Decision Trees and a Genetic Algorithm Wrapper. *Informatics* **2022**, *9*, 45. [CrossRef]
42. Mabdeh, A.N.; Al-Fugara, A.; Ahmadlou, M.; Al-Adamat, R.; Al Shabeeb, A.R. GIS-based landslide susceptibility assessment and mapping in Ajloun and Jerash governorates in Jordan using genetic algorithm-based ensemble models. *Acta Geophys.* **2022**, *70*, 1253–1267. [CrossRef]
43. Tao, P.; Sun, Z.; Sun, Z. An Improved Intrusion Detection Algorithm Based on GA and SVM. *IEEE Access* **2018**, *6*, 13624–13631. [CrossRef]
44. Kasongo, S.M. An Advanced Intrusion Detection System for IIoT Based on GA and Tree Based Algorithms. *IEEE Access* **2021**, *9*, 113199–113212. [CrossRef]
45. Credit Card Fraud Detection. Available online: https://kaggle.com/mlg-ulb/creditcardfraud (accessed on 26 October 2021).
46. Lin, T.-H.; Jiang, J.-R. Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest. *Mathematics* **2021**, *9*, 2683. [CrossRef]
47. Mienye, I.D.; Obaido, G.; Aruleba, K.; Dada, O.A. Enhanced Prediction of Chronic Kidney Disease Using Feature Selection and Boosted Classifiers. In *Intelligent Systems Design and Applications*; Springer: Cham, Switzerland, 2022; pp. 527–537. [CrossRef]
48. Alhaj, T.A.; Siraj, M.M.; Zainal, A.; Elshoush, H.T.; Elhaj, F. Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation. *PLoS ONE* **2016**, *11*, e0166017. [CrossRef] [PubMed]
49. Ebiaredoh-Mienye, S.A.; Swart, T.G.; Esenogho, E.; Mienye, I.D. A Machine Learning Method with Filter-Based Feature Selection for Improved Prediction of Chronic Kidney Disease. *Bioengineering* **2022**, *9*, 350. [CrossRef]
50. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [CrossRef]
51. Schulte, R.V.; Prinsen, E.C.; Hermens, H.J.; Buurke, J.H. Genetic Algorithm for Feature Selection in Lower Limb Pattern Recognition. *Front. Robot. AI* **2021**, *8*, 710806. Available online: https://www.frontiersin.org/articles/10.3389/frobt.2021.710806 (accessed on 23 November 2022). [CrossRef] [PubMed]
52. Kalita, K.; Dey, P.; Haldar, S.; Gao, X.-Z. Optimizing frequencies of skew composite laminates with metaheuristic algorithms. *Eng. Comput.* **2020**, *36*, 741–761. [CrossRef]
53. Jovanovic, D.; Antonijevic, M.; Stankovic, M.; Zivkovic, M.; Tanaskovic, M.; Bacanin, N. Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection. *Mathematics* **2022**, *10*, 2272. [CrossRef]
54. Prasetiyowati, M.I.; Maulidevi, N.U.; Surendro, K. Determining threshold value on information gain feature selection to increase speed and prediction accuracy of random forest. *J. Big Data* **2021**, *8*, 84. [CrossRef]
55. Xie, J.; Wang, M.; Xu, S.; Huang, Z.; Grant, P.W. The Unsupervised Feature Selection Algorithms Based on Standard Deviation and Cosine Similarity for Genomic Data Analysis. *Front. Genet.* **2021**, *12*, 684100. Available online: https://www.frontiersin.org/article/10.3389/fgene.2021.684100 (accessed on 15 January 2022) [CrossRef]
56. Van Hulse, J.; Khoshgoftaar, T.M.; Napolitano, A.; Wald, R. Threshold-based feature selection techniques for high-dimensional bioinformatics data. *Netw. Model. Anal. Health Inform. Bioinform.* **2012**, *1*, 47–61. [CrossRef]

57. Theodoridis, P.K.; Gkikas, D.C. Optimal Feature Selection for Decision Trees Induction Using a Genetic Algorithm Wrapper—A Model Approach. In *Strategic Innovative Marketing and Tourism*; Springer: Cham, Switzerland, 2020; pp. 583–591. [CrossRef]
58. Kumar, A.; Sinha, N.; Bhardwaj, A. A novel fitness function in genetic programming for medical data classification. *J. Biomed. Inform.* **2020**, *112*, 103623. [CrossRef]
59. Mienye, I.D.; Sun, Y. Effective Feature Selection for Improved Prediction of Heart Disease. In *Pan-African Artificial Intelligence and Smart Systems*; Springer: Cham, Switzerland, 2022; pp. 94–107. [CrossRef]
60. Costa-Carrapiço, I.; Raslan, R.; González, J.N. A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. *Energy Build.* **2020**, *210*, 109690. [CrossRef]
61. Maghawry, A.; Hodhod, R.; Omar, Y.; Kholief, M. An approach for optimizing multi-objective problems using hybrid genetic algorithms. *Soft Comput.* **2021**, *25*, 389–405. [CrossRef]
62. Blank, J.; Deb, K. A Running Performance Metric and Termination Criterion for Evaluating Evolutionary Multi- and Many-objective Optimization Algorithms. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
63. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
64. Schapire, R.E. A brief introduction to boosting. *IJCAI* **1999**, *99*, 1401–1406.
65. Cramer, J.S. The Origins of Logistic Regression. In *Social Science Research Network*; SSRN Scholarly Paper ID 360300; SSRN: Rochester, NY, USA, 2002. [CrossRef]
66. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
67. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
68. Krzywinski, M.; Altman, N. Classification and regression trees. *Nat. Methods* **2017**, *14*, 8. [CrossRef]
69. Prusty, S.; Patnaik, S.; Dash, S.K. SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Front. Nanotechnol.* **2022**, *4*, 972421. Available online: https://www.frontiersin.org/articles/10.3389/fnano.2022.972421 (accessed on 8 November 2022). [CrossRef]
70. Trevethan, R. Sensitivity, Specificity, and Predictive Values: Foundations, Pliabilities, and Pitfalls in Research and Practice. *Front. Public Health* **2017**, *5*, 307. Available online: https://www.frontiersin.org/article/10.3389/fpubh.2017.00307 (accessed on 25 January 2022). [CrossRef]
71. Mienye, I.D.; Sun, Y. A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access* **2022**, *10*, 99129–99149. [CrossRef]
72. Obaido, G.; Ogbuokiri, B.; Swart, T.G.; Ayawei, N.; Kasongo, S.M.; Aruleba, K.; Mienye, I.D.; Aruleba, I.; Chukwu, W.; Osaye, F.; et al. An Interpretable Machine Learning Approach for Hepatitis B Diagnosis. *Appl. Sci.* **2022**, *12*, 11127. [CrossRef]
73. Mienye, I.D.; Sun, Y.; Wang, Z. Improved Predictive Sparse Decomposition Method with Densenet for Prediction of Lung Cancer. *Int. J. Comput.* **2020**, *1*, 533–541. [CrossRef]
74. Zain, A.M.; Haron, H.; Sharif, S. Application of GA to optimize cutting conditions for minimizing surface roughness in end milling machining process. *Expert Syst. Appl.* **2010**, *37*, 4650–4659. [CrossRef]
75. Mirjalili, S. Genetic Algorithm. In *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Mirjalili, S., Ed.; Springer International Publishing: Cham, Switzerland, 2019; pp. 43–55. [CrossRef]
76. Mienye, I.D.; Kenneth Ainah, P.; Emmanuel, I.D.; Esenogho, E. Sparse noise minimization in image classification using Genetic Algorithm and DenseNet. In Proceedings of the 2021 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 10–11 March 2021; pp. 103–108. [CrossRef]
77. Zhu, H.; Liu, G.; Zhou, M.; Xie, Y.; Abusorrah, A.; Kang, Q. Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection. *Neurocomputing* **2020**, *407*, 50–62. [CrossRef]
78. Alkhatib, K.I.; Al-Aiad, A.I.; Almahmoud, M.H.; Elayan, O.N. Credit Card Fraud Detection Based on Deep Neural Network Approach. In Proceedings of the 2021 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 24–26 May 2021; pp. 153–156. [CrossRef]
79. Yotsawat, W.; Wattuya, P.; Srivihok, A. A Novel Method for Credit Scoring Based on Cost-Sensitive Neural Network Ensemble. *IEEE Access* **2021**, *9*, 78521–78537. [CrossRef]
80. Kalid, S.N.; Ng, K.-H.; Tong, G.-K.; Khor, K.-C. A Multiple Classifiers System for Anomaly Detection in Credit Card Data With Unbalanced and Overlapped Classes. *IEEE Access* **2020**, *8*, 28210–28221. [CrossRef]
81. Mrozek, P.; Panneerselvam, J.; Bagdasar, O. Efficient Resampling for Fraud Detection During Anonymised Credit Card Transactions with Unbalanced Datasets. In Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), Leicester, UK, 7–10 December 2020; pp. 426–433. [CrossRef]
82. Carta, S.; Ferreira, A.; Reforgiato Recupero, D.; Saia, R. Credit scoring by leveraging an ensemble stochastic criterion in a transformed feature space. *Prog. Artif. Intell.* **2021**, *10*, 417–432. [CrossRef]
83. Xie, Y.; Li, A.; Gao, L.; Liu, Z. A Heterogeneous Ensemble Learning Model Based on Data Distribution for Credit Card Fraud Detection. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, e2531210. [CrossRef]
84. Saheed, Y.K.; Hambali, M.A.; Arowolo, M.O.; Olasupo, Y.A. Application of GA Feature Selection on Naive Bayes, Random Forest and SVM for Credit Card Fraud Detection. In Proceedings of the 2020 International Conference on Decision Aid Sciences and Application (DASA), Sakheer, Bahrain, 8–9 November 2020; pp. 1091–1097. [CrossRef]

85. Verma, B.P.; Verma, V.; Badholia, A. Hyper-Tuned Ensemble Machine Learning Model for Credit Card Fraud Detection. In Proceedings of the 2022 International Conference on Inventive Computation Technologies (ICICT),Nepal, 20–22 July 2022; pp. 320–327. [CrossRef]
86. Padhi, B.K.; Chakravarty, S.; Naik, B.; Pattanayak, R.M.; Das, H. RHSOFS: Feature Selection Using the Rock Hyrax Swarm Optimization Algorithm for Credit Card Fraud Detection System. *Sensors* **2022**, *22*, 9321. [CrossRef]
87. Ganji, V.R.; Chaparala, A.; Sajja, R. Shuffled shepherd political optimization-based deep learning method for credit card fraud detection. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7666. [CrossRef]
88. UCI Machine Learning Repository: Statlog (German Credit Data) Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data) (accessed on 5 December 2022).
89. UCI Machine Learning Repository: Default of credit card clients Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients (accessed on 5 December 2022).

CREDIT CARD FRAUD DETECTION

Rachitha E[1], Rani H E[2], Prathiksha[3], Swathi B V[4],

[1] *Student, Information Science & Engineering, SJB Institute of Technology, Karnataka, India*
[2] *Student, Information Science & Engineering, SJB Institute of Technology, Karnataka, India*
[3] *Student, Information Science & Engineering, SJB Institute of Technology, Karnataka, India*
[4] *Assistant Professor, Information Science & Engineering, SJB Institute of Technology, Karnataka, India*

ABSTRACT

In recent days the use of credit card for all types of transactions has been increased drastically all over the globe which indicates the increase in fraudulent transaction. A large number of fraudulent transactions are made each and every second across the global where transactions can be determined using various modern techniques like data mining, machine learning and few others which have been used recently. This paper uses various genetic algorithms for finding accurate solutions for the modern day problems like detecting the unusual transactions. As the number of online transactions are increasing day by day the responsibilities of the banks to increase the security to these transactions. Our proposed system focuses on series of machine learning models where we select the best method among these available algorithms.

Keywords*: machine learning, Data mining*

1. INTRODUCTION

A credit card is a thin card which is made of plastic and is handy, it contains information such as signature are picture for the identification purpose. It gives the authorization to the person to whom it belongs, for the purpose of purchasing and charge services to his /her account (charges for which he will be billed periodically). The information on the credit card can be read by automated teller machines, store readers ,bank and can also be used in online banking systems. The security of the credit card depends on the physical security of the plastic card as well as the privacy of the credit card number.

A fraud is detected when one individual uses the other individual's card for their own use without intimating the owner about the transaction. During this kind of transactions, the fraudsters can use the card until its available time limit is depleted. Therefore, a solution which reduces the total available limit on the credit card should be found.

This type of fraud is done when any person with pure intensions to defraud uses card of unknown which has been lost, stolen, cancelled or revoked and misuses which results in fraud. Using number of credit card without having actual card can be also known as credit card fraud. Identity theft also have been increasing and have contributed to fraudulent transaction. This effects consumer credit industry as it has become one of the fastest growing type and which is most complicated and difficult in solving.

2. PROPOSED SYSTEM

This proposed system uses the PCA which means principal component analysis where these principal components are computed which are further used to perform the change of the basis on the data available. In our paper, we have opposed PCA on the dataset for cleaning the data and for reducing the data.

During the data pre-processing, data will be cleaned, integrated, transformed and reduced to the required form. The missing data is handled by applying data handling techniques. The number of fraudulent transactions is always less when compared to the number of genuine transactions. To overcome this, we have used Random over-sampling

techniques where the minority occurrences are raised. This pre-processed data set will be trained first and then tested.

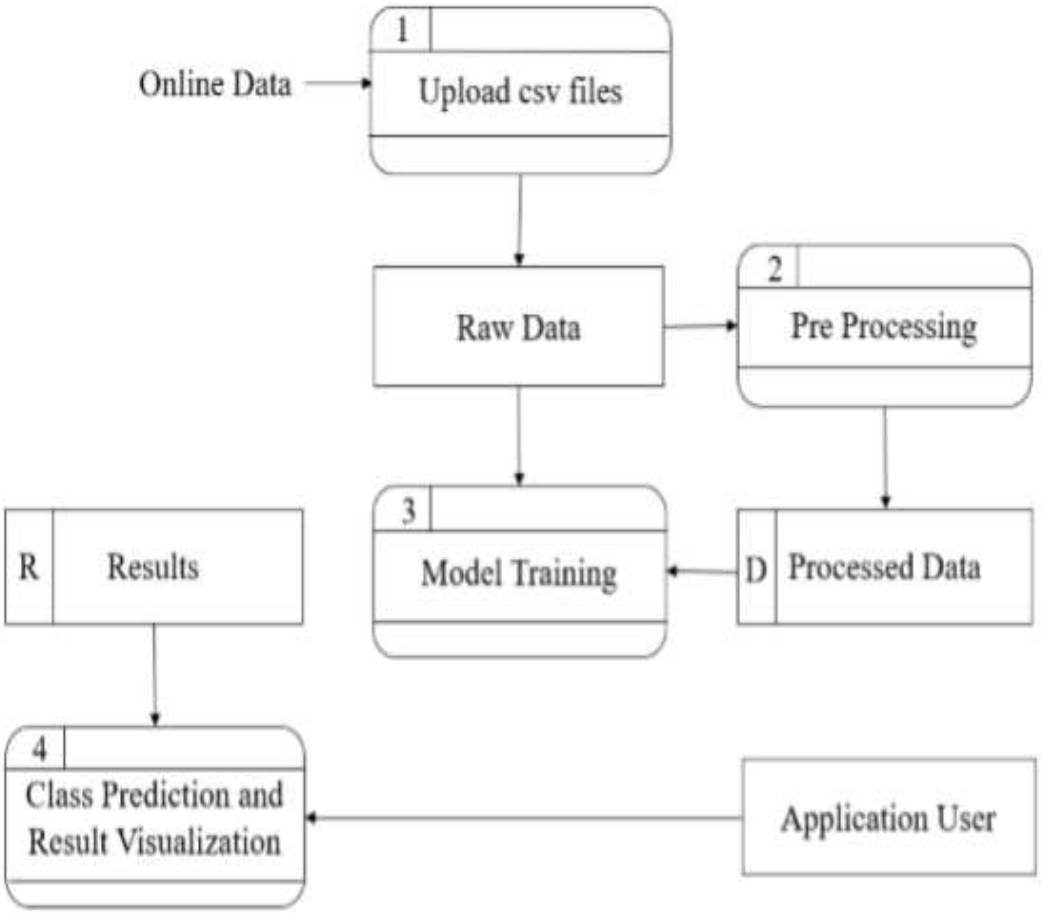


Fig -1: Flow diagram

2.1 Random Forest Algorithm

Random forest algorithm for finding the fraudulent transactions and the accuracy of these transaction and it is based on supervised learning algorithm where it uses decision trees for classification of the dataset and also handle the missing values.

2.2 Logistic Regression

It is a supervised learning used for visualization purpose also mainly used for classification of data and predict the target variable. Even though, it is named as regression this algorithm is used for classification purpose.

2.3 Naïve Bayes Algorithm

The naïve Bayes classifier is the probability-based classifier for the fraud detection. Here, the target classes probability and test case probability are calculated. Naïve **B**ayes classifier is based on applying the Bayes theorem with strong assumptions between the features and the different parameters that are computed such as, precession, recall and sensitivity.

2.4 Decision Algorithm

Decision tree is a supervised learning technique. In this paper, it has been used for classifying of the data. It is used to create a training model, which is then used to predict the value or the class of target variable while learning simple decision rules which are taken from prior data.

3. RESULTS

In credit card fraud detection, after using random forest, naïve bayes, logistic regression, decision tree algorithm, the most accuracy seen is the random forest with 99.6% of accuracy which was the best one when compared to the accuracy of the rest followed by the least accuracy with 97.2% which is of naïve bayes.

Over all with the practical accuracy results found random forest suited the best with best accuracy compared to another algorithm**.**

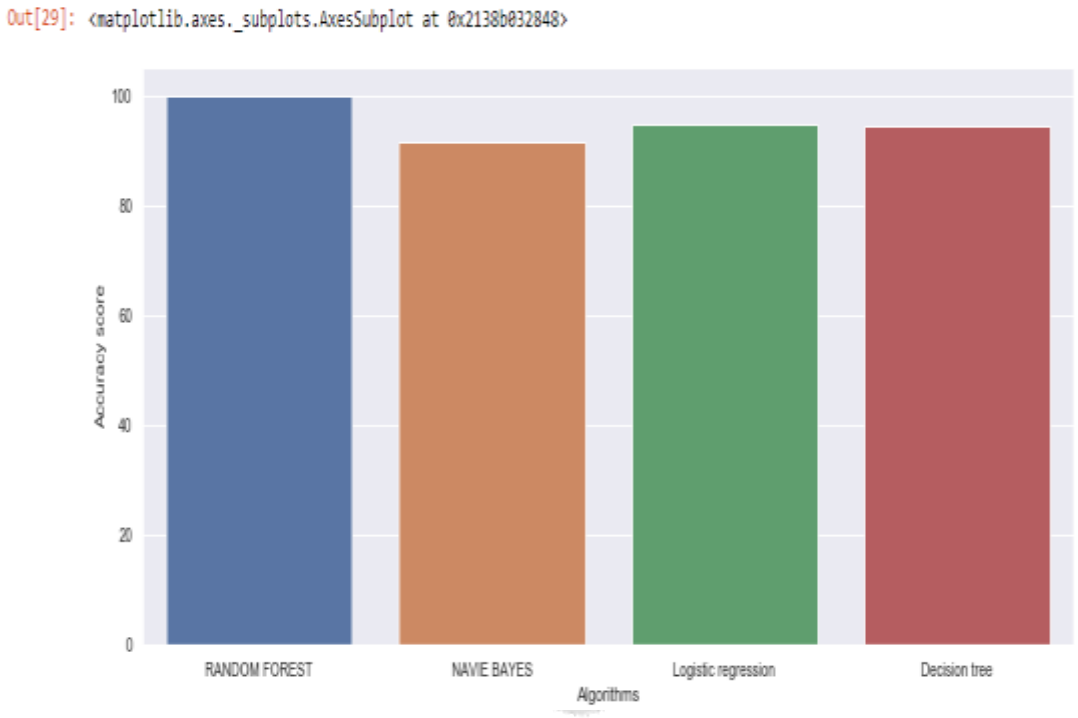


Fig - 2: Graphical Representation of Algorithms

Considering the obtained results, Random forest algorithm can be considered are the best algorithm which gives the best accuracy. Hence this algorithm can be used.

4. CONCLUSION AND FUTURE ENHANCEMENT

One of the common online criminal activities are the credit card fraudulent activities. This paper helps in the detection of fraudulent activities happening during online credit card transactions. here we have used random forest algorithm for the detection of fraudulent transactions. The random forest algorithm gives the accuracy around 99.6%, The below figure shows the graphical representation of the accuracies obtained when the following algorithms are applied to the dataset.

5. REFERENCES

1] S. Xuan, G. Liu, Z. Li, L. Zheng, S. Wang, and G. N. Surname, **"**Random forest for credit card fraud detection**",** IEEE 15th International Conference on Networking, Sensing and Control (ICNSC),2018.
[2] Dilip Singh Sisodia, Nerella Keerthana Reddy, Shivangi Bhandari, "Performance Evaluation of Class Balancing Techniques for Credit Card Fraud Detection" IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017).
[3] SamanehSorournejad , Zahra Zojaji , Reza Ebrahimi Atani , Amir Hassan Monadjemi, "A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective",IEEE 2016
[4] E. Michael and S. Pedro, "A survey of signature-based methods for financial fraud detection," *Computer and security*, vol. vol 28, no. 6, pp. 381–394.
[5] B. Adrian, "Detecting and Preventing Fraud with Data Analytics," *Procedia Economics and Finance*, vol. 32, no. 15, pp. 1827–1836, 2015.
[6] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
[7] B. Zhu, B. Baesens, and K. L. M. Seppe, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information Sciences*, vol. 408, pp. 84–99, 2017.
[8]N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
[9] G. E. a. P. a. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017)
[10] Sailusha, R., Gnaneswar, V., Ramesh, R., & Rao, G. R. (2020). *Credit Card Fraud Detection Using Machine Learning. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS).*

Credit Card Fraud Detection Using Bayesian and Neural Networks

Sam Maes Karl Tuyls Bram Vanschoenwinkel
Bernard Manderick
Vrije Universiteit Brussel - Department of Computer Science
Computational Modeling Lab (COMO)
Pleinlaan 2
B-1050 Brussel, Belgium
{sammaes@,ktuyls@,bvschoen@,bernard@arti.}vub.ac.be

Abstract

This paper discusses automated credit card fraud detection by means of machine learning. In an era of digitalization, credit card fraud detection is of great importance to financial institutions. We apply two machine learning techniques suited for reasoning under uncertainty: artificial neural networks and Bayesian belief networks to the problem and show their significant results on real world financial data. Finally, future directions are indicated to improve both techniques and results.

1 Introduction

A marking example of the digitalization of our society over the last years is the proliferation of credit card use. This evolution did also bring forth the problem of credit card fraud, where ever more sophisticated methods are used to steal considerable amounts of money. This paper discusses the problem of identifying or detecting fraudulent behavior in a credit card transaction system.

Reasoning under uncertainty is a key element in many artificial intelligence applictions in general and machine learning in particular. Therefore, many formalisms have been developed that support that kind of reasoning: probabilistic reasoning, fuzzy logic, etc. Here, we focus on two machine learning techniques to the credit card fraude detections problem: Artificial Neural Networks (ANNs) and Bayesian Belief Networks (BBNs). The central idea is to provide some computational learner with a set of training data consisting of some feature values (e.g. extracted from the data of a series of financial transactions) that are inherent to the system in which we want to do the fraud detection. After a process of learning, the program is supposed to be able to correctly classify a transaction it has never seen before as fraudulent or not fraudulent, given some features of that transaction.

The structure of this paper is as follows: first we introduce the reader to the domain of credit card fraud detection. In Sections 3 and 4 we briefly explain the two Machine Learning techniques used, respectively ANN and BBN. Finally, we discuss some of the experiments conducted for both techniques, followed by a conclusion and possible future directions of research.

2 Credit Card Fraud Detection

In this section we discuss credit card fraud and the specific problems that arise with it. We will focus on credit card fraud, but most properties also apply to other real world problems, such as cellular phone fraud, calling card fraud and computer network intrusion.

2.1 What is Credit Card Fraud ?

The advent of credit cards and their increasing functionality have not only given people more personal comfort, but have also attracted malicious characters interested in the handsome rewards to be earned.

Credit cards are a nice target for fraud, since in a very short time a lot of money can be earned without taking to many risks. This is because often the crime is only discovered a few weeks after date.

Some successful credit card fraud techniques are

- Copying a credit card and in some way getting hold of the secret pin-code of the user (if

needed).

- Vendors charging more money than agreed to the customer, without the latter being aware of it.

When banks lose money due to credit card fraud, card holders partially (possibly entirely) pay for the loss through higher interest rates, higher membership fees, and reduced benefits. Hence, it is both the banks' and card holders' interest to reduce illegitimate use of credit cards and that is the reason why financial institutions started to do fraud detection.

Fraud detection is, given a set of credit card transactions, the process of identifying those transactions that are fraudulent, i.e., classifying the transactions into two classes: a class of genuine and a class of fraudulent transactions.

2.2 Problems with Credit Card Fraud Detection

One of the biggest problems associated with fraud detection is the lack of both literature providing experimental results and of real world data for academic researchers to perform experiments on. This is because fraud detection is often associated with sensitive financial data that is kept confidential for reasons of customer privacy.

We now enumerate some of the properties a fraud detection system should have in order to perform good results.

- The system should be able to handle *skewed distributions*, since only a very small percentage of all credit card transactions is fraudulent. To solve this problem, often the training sets are divided into pieces where the distribution is less skewed [Chan98].

- The ability to handle *noise*. This is simply the presence of errors in the data, for instance incorrect dates. Noise in actual data limits the accuracy of generalization that can be achieved, no matter how extensive the training set is.

  One way to deal with this problem is by cleaning the data [Faw97].

- *Overlapping data* is another problem in this field. Many transactions may resemble fraudulent transactions, when actually they are legitimate. The opposite also happens, when a fraudulent transaction appears to be normal.

- The systems should be able to *adapt* themselves to new kinds of fraud. Since after a while successful fraud techniques decrease in efficiency, due to the fact that they become well known. Then a "good" fraud tries to find new and inventive ways of doing his job.

- There is a need for *good metrics* to evaluate the classifier system. As an example, the overall accuracy is not suited for evaluation on a skewed distribution, since even with a very high accuracy, almost all fraudulent transactions can be misclassified.

- The systems should take into account the *cost* of the fraudulent behavior detected and the cost associated with stopping it. For example, no profit is made by stopping a fraudulent transaction of only a few Euros.

  This means that there should be a *decision layer* on top of the fraud detection system. The decision layer decides what action to take when fraudulent behavior is detected via the fraud detection system, taking into account factors like the amount of the transaction and the quality of the customer doing the transaction.

3 Artificial Neural Networks

3.1 Definition

Neural Networks exist in many ways and different forms. The type of neural network we will discuss in this paper is the Feed Forward Multi-layer Perceptron. A feed forward multi-layer perceptron consists of different layers of perceptrons that are interconnected by a set of weighted connections. We can distinguish three types of layers:

- Input layer: Receives input from an input stream, which can be a database or some device or something else.

- Hidden layer: Is hidden from the outside world and receives input only from the input layer or another hidden layer.

- Output layer: Connects the network to the outside world again and provides the final output of the network.

A feed forward multi-layer perceptron has no cycles and there is full connectivity between the perceptrons of two consecutive layers. Signals can be propagated in two directions: function signals are propagated forwards, i.e. from input layer through the

hidden layer(s) to the output layer and error signals are propagated backwards, i.e. from output layer through the hidden layer(s) to the input layer.

3.2 Learning

The type of learning we will discuss is commonly called supervised learning or error correction learning. The algorithm that we have used to do this is called **Backpropagation of Error Signals** or in short Backprop. Every iteration of the algorithm consists of two passes [1]:

1. **Forward pass**: every perceptron calculates the weighted linear combination of all its inputs and applies to the result of this summing junction an activation function. The result of the activation function provides the perceptron of its output value.

2. **Backward pass**: At the output layer of the network we calculate the error with respect to the desired output value for a certain pattern. This error is propagated backwards through the network enforcing a correction on the weights of all connections in the network. This technique is based on the observation that all perceptrons in the network have a shared responsibility for the error that has been calculated at the output layer.

For more details concerning Backprop and other issues concerning learning in a feed forward multi-layer perceptron we refer to [Mae00].

4 Bayesian Networks

4.1 Definition

A Bayesian network is a directed acyclic graph that consists of a set of random variables. Each variable has a finite set of mutually exclusive states. A set of directed links or arrows connects pairs of nodes. The intuitive meaning of an arrow from node X to node Y is that X has a direct **influence** on Y.

A Bayesian network represents the dependence between the variables and gives a compact specification of the joint probability distribution. In fact a BBN is a factorization of the joint probability. Each node of the network has a conditional probability table (CPT) that quantifies the effect of the parent nodes.

[1] At every iteration of the algorithm we present the network with a certain pattern taken from a training set. The features of the pattern are presented to the different perceptrons at the input layer of the network.

The parents of a node are all those nodes that have arrows pointing to it. For orphan nodes this reduces to prior probabilities.

4.2 Learning

This section deals with the problem of constructing a network automatically from direct empirical observations.

Taking Bayesian belief networks as the basic scheme of knowledge representation, the learning task separates into two additional subtasks:

1. Identifying the topology of the network, specifically, the missing links and the directionality of the arrows.

2. Learning the numerical parameters (the prior and conditional probabilities) for a given network topology.

Since the second task is trivial given enough data, we will only concentrate ourselves on learning the topology of the network. There are several approaches to do this, such as dependency analysis [Chen97] and global optimization. We opted for STAGE [Boy98] which is an instance of the latter.

Global optimization is the problem of finding the best possible configuration from a large space of possible configurations.

Formally, an instance of such a global optimization consists of a *state space* X and an *objective function* Obj: $X \rightarrow \Re$. This objective function evaluates the quality of the state as a final solution, by transforming the state into a real number. The goal of global optimization is to find a state $x^* \in X$, which minimizes Obj, that is, $\text{Obj}(x^*) \leq \text{Obj}(x)\ \forall x \in X$. If the space X is so small that every state can be evaluated, obtaining the exact solution x^* is trivial, otherwise, special knowledge of the problem structure must be exploited.

The STAGE algorithm aims to exploit the following observation: the performance of a local search algorithm depends on the state from which the search starts. We express this as follows:

The value function $V^\pi(x) \equiv$ expected best Obj value seen on a trajectory that starts from state x and follows local search method π. Intuitively $V^\pi(x)$ evaluates x's *promise* as a starting state for π.

We seek to approximate V^π using a function approximation model such as linear regression or multilayer perceptrons, where states x are encoded as real-valued feature vectors. These input features may encode any relevant properties of the state, including the original objective function $\text{Obj}(x)$ itself.

We denote the mapping from states to features by $F : X \rightarrow \Re^D$, and our approximation of $V^\pi(x)$ by $\widetilde{V}^\pi(F(x))$.

Training data for supervised learning of $\widetilde{V}^\pi$ may be readily obtained by running π from different starting points. Moreover, if the algorithm π behaves as a Markov chain, intermediate states of each simulated trajectory may also be considered alternate "starting points" for that search, and thus used as training data for $\widetilde{V}^\pi$ as well. This insight enables us to get not one but perhaps hundreds of pieces of training data from each trajectory sampled.

The learned evaluation function $\widetilde{V}^\pi(F(x))$ evaluates how promising x is as a starting point for local search algorithm π. To find the best starting point, we must optimize $\widetilde{V}^\pi$ over X. We do this by simply applying stochastic hill-climbing with $\widetilde{V}^\pi$ instead of Obj as the evaluation function.

To summarize, STAGE repeatedly alternates between two different stages of local search: running the original local search method π on Obj, and running hill-climbing on $\widetilde{V}^\pi$ to find a promising new starting state for π. Thus, STAGE can be viewed as a *smart multi-restart* approach to local search. The operation of STAGE is schematically depicted in Figure 1.

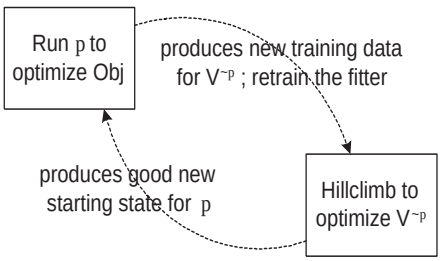


Figure 1: STAGE alternates between optimizing Obj with π and hill-climbing to optimize $\widetilde{V}^\pi$.

4.3 Stage applied to Bayesian networks

In this part we will apply the STAGE approach from the previous subsection to the problem of Bayesian network structure learning.

First we need an objective function Obj which must return a value that quantifies the quality of a Bayesian network.

We will use the Minimum Description Length (MDL) metric [Lam94], which trades off between maximizing fit accuracy and minimizing model complexity. Consider a network x with A nodes. The objective function decomposes into a sum over the nodes of the network x:

$$Obj(x) = \sum_{j=1}^{A} (-Fitness(x_j) + K \cdot Complexity(x_j)) \tag{1}$$

The *Fitness* term computes a mutual information score at each node x_j by summing over all possible joint assignments to variable j and its parents:

$$Fitness(x_j) = \sum_{v_j} \sum_{V_{Par_j}} N(v_j \wedge V_{Par_j}) \log \frac{N(v_j \wedge V_{Par_j})}{N(V_{Par_j})} \tag{2}$$

Here, $N(\cdot)$ refers to the number of records in the database that match the specified variable assignment.

The *Complexity* term simply counts the number of parameters required to store the conditional probability table at node j:

$$Complexity(x_j) = (Arity(j) - 1) \prod_{i \in Par_j} Arity(i), \tag{3}$$

where the *Arity* of a node in a Bayesian network is defined as the number of states the variable associated with the node can adopt.

The constant K in 1 is set to $log(R)/2$, where R is the number of records in the database.

Including the *Complexity* term from 3 into the objective function from 1 introduces Occam's razor in a natural way: since both K and the *Complexity* term are always positive values and since we want to minimize the objective function, there will be a tendency towards simple structures.

The local search method π that we will use is stochastic hill-climbing. Because directed cyclic graphs are not allowed in Bayesian networks we have to ensure that the graphs visited in π are acyclic. This is done by maintaining a permutation $x_{i_1}, x_{i_2}, \ldots, x_{i_A}$ on the A nodes, and all links in the graph are directed from nodes of lower index to nodes of higher index. Local search begins from a linkless graph on the identity permutation after which the following move operators are used: [2]

- With probability 0.7, choose two random nodes of the network and add a link between them (if

[2] Again, Occam's razor applies, since the simplest structures are evaluated first, followed by more complex structures.

that link isn't already there) or delete the link between them (otherwise).

- With probability 0.3, swap the permutation ordering of two random nodes of the network. This may cause multiple graph edges to be reversed, namely those that don't point from nodes with a lower index to those with a higher index after the permutation.

For learning, STAGE was given the following seven extra features:

- Features 1-2: mean and standard deviation of *Fitness* over all the nodes
- Features 3-4: mean and standard deviation of *Complexity* over all the nodes
- Features 5-6: mean and standard deviation of the number of parents of each node
- Feature 7: the number of "orphan" nodes, i.e., nodes that don't have parents

The fitter used to train $\widetilde{V}^{\pi}$ can be any method to do function approximation, like linear or quadratic regression. We used feedforward multi-layer perceptron in our implementation.

5 Experiments

5.1 Methodology

In this subsection we present a methodology to describe an experiment and its results.

The data we use is real world data that has been provided to us by Serge Waterschoot at Europay International (EPI). This data consists of a set of features that contain useful information about a transaction, we will label these features by F_i without specifying them. Unfortunately, we cannot specify them, because the agreement with EPI forbids us to do so.

We introduce a measure of performance that is independent of the learning problem at stake and gives us a clear idea about the quality of a result. For this purpose we introduce the Receiver Operating Curve (ROC) (see fig 1).

After the training of a network (ANN or BBN) it will be applied to a set of features it has never seen before. Of course, we can say something about the classification of the transactions in this set, for example, how many of these transactions have been correctly classified as genuine. Or even better, how much percent of the total of these transactions have been classified correctly as genuine.

We are especially interested in knowing how many fraudulent transactions are classified correctly as fraudulent and at the same time how many genuine transactions are classified incorrectly as fraudulent. The first is called the true positive rate and the latter the false positive rate. The ROC will combine this information in one graph. We will plot the false positives (x-axis) against the true positives (y-axis). By doing this, we get a concave shaped graph that contains for each data point on this graph the following information:

- on the x-axis you can read the percentage of transactions that have been classified incorrectly as fraudulent
- on the y-axis you can read the percentage of transactions that have been classified correctly as fraudulent

The steeper the increase with respect to the y-axis, while the values on the x-axis remain small, the better there has been learned, the better the predictions. The line, dividing the first quadrant, called the bisection, corresponds to a model that has done no learning.

5.2 Fraud Detection with Artificial Neural Networks

It is often wrongly assumed that neural networks are a fast, easy and reliable technique to obtain good results in different areas. In practice it is found that the great difficulty in applying neural networks resides in the choice of a good set of pre-processing operations and a good trade-off between the different parameters that have to be chosen.

The first experiment [3] shows the importance of pre-processing. In figure 2 (a) you can see two ROC curves, the best result in this graph has been obtained by performing a correlation analysis on the 10 features of the original data set. This resulted in the observation that one feature was strongly correlated with many of the other features. Removal of this feature clearly improves the results. For clarity we will point out some of the results on the ROC curves:

1. Dark ROC, is the best result, pre-processing: normalization, desired values are offset ϵ away

[3] For all experiments we used: a training set to train the network, a test set to calculate the average mean square error over the perceptrons at the output layer and a validation set to produce the ROC curves.

from the real desired values and correlation analysis (resulting in the removal of one feature): for 70 percent true positives we have only 15 percent false positives.

2. Light ROC, pre-processing: normalization, desired values are offset ϵ away from the real desired values: for 60 percent true positives we already have 15 percent false positives.

The second experiment shows the influence of parameter tuning on the process of learning. By decreasing the learning rate at certain intervals of the learning process we can improve the speed and efficiency of this process. This is illustrated in figure 2(b). Figure 2 (c) shows the corresponding ROC curve of this experiment. The learning rate has been dropped at epochs 5, 10, 30, 57.

5.3 Fraud Detection with Bayesian Belief Networks

We conducted one experiment on a dataset where each transaction is described by 4 features and a fraud label. A structure that received a high score from the STAGE algorithm can be seen in Figure 2 (d). As you can see two features influence fraud and fraud influences two features. Figure 2 (e) depicts the ROC associated with this structure and we can see that it performs well. For example, when 68% of the fraudulent transactions are correctly recognized, then only 10% of the genuine transactions are falsely classified as fraudulent.

Another experiment was conducted on a dataset where each transaction is described by 10 features and a fraud label. One structure returned by the learning algorithm can be seen in Figure 2 (f), and its matching ROC in Figure 2 (g). On the ROC we see that when we allow 15% of the fraudulent transactions to be incorrectly classified, 73% of the fraudulent transactions.

5.4 Comparison

- The results of BBN and ANN are compared in table 1. We can see that BBN performs better than ANN applied to fraud detection. For instance comparing ANN-(a) to BBN-(e) in the table, you can see differences of approximately 8%. This means that in some cases, BBN detects 8% more of the fraudulent transactions.

- Learning times can go to several hours for ANN and take up to 20 minutes for BBN.

- The evaluation of new examples is typically much faster for ANN than for BBN.

experiment	±10% false pos	±15% false pos
ANN-fig 2(a)	60% true pos	70% true pos
ANN-fig 2(a)	47% true pos	58% true pos
ANN-fig 2(c)	60% true pos	70% true pos
BBN-fig 2(e)	68% true pos	74% true pos
BBN-fig 2(g)	68% true pos	74% true pos

Table 1: This table compares the results achieved with ANN and BBN, for a false positive rate of respectively 10% and 15%.

6 Conclusion

In this paper we showed that good results can be achieved by applying ANN and BBN to fraud detection. As the comparison shows Bayesian Networks yield better results concerning fraud detection and their training period is shorter but the fraud detection process is considerably faster with ANNs.

Finally we point out what interesting ongoing work we are still doing and what extensions could be added to our implementations of both ANN and BBN.

Future work ANN

- Pruning algorithms exist to cut away connections and perceptrons that are practically not used during training, this can significantly improve the performance of backprop.

- The use of variations, like radial basis networks, support vector machines (SVMs) and backprop performing weight updates with respect to some other error function.

Future work BBN

- Extend the implementation so that continuous variables are allowed in the networks.

- Implement a structure learning method that is based on dependency analysis (as opposed to the search & scoring method, as in STAGE) and compare the results with those of the STAGE algorithm.

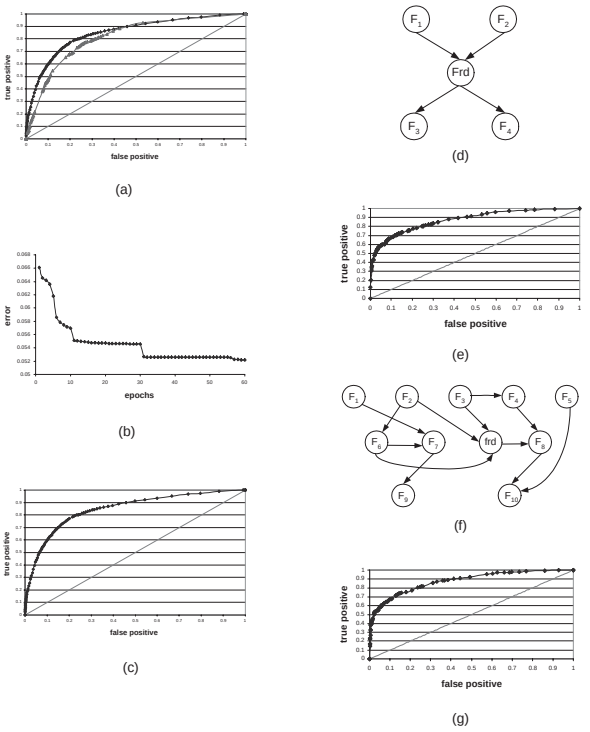


Figure 2: (a) Two ROC curves: The best result is obtained by performing a correlation analysis. (b)Average mean square error: every epoch is a complete pass of the training set through the network. (c) ROC curve corresponding to the error curve in (b). (d) A structure of five features, which received a high score from the STAGE algorithm. (e) The ROC associated with (d). (f) A structure of ten features, which received a high score from the STAGE algorithm. (g) The ROC associated with (f).

References

[Bis96] Bishop, C.M., Neural Networks for pattern recognition. Oxford University Press, 1996.

[Boy98] Boyan, J.A., Learning evaluation functions for global optimization, 1998.

[Chan97] Chan, P.K., Stolfo, S.J., Fan, D.W., Lee, W., Prodromidis, A.L., Credit card fraud detection using meta learning: Issues and initial results. Working notes of AAAI Workshop on AI Approaches to Fraud Detection and Risk Management, 1997.

[Chan98] Chan, P.K., Stolfo, S.J., Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 164–168, 1998.

[Chen97] Cheng, J., Bell, D.A. and Liu, W., Learning Bayesian networks from data: An efficient approach based on information theory. In Proceedings of ACM CIKM'97.

[Dou95] Dougherty, J., Kohavi, R., Sahami, M., Supervised and unsupervised discretization of continuous features. Proceedings of the Twelfth International Conference on Machine Learning (pp. 194–202). Tahoe City, CA: Morgan Kaufmann, 1995.

[Faw97] Fawcett, T., Provost, F., Adaptive Fraud Detection. Data Mining and Knowledge Discovery, 1(3), 1997.

[Fay97] Fayyad, U.M., Mannila, H., Piatetsky-Shapiro, G., Data mining and knowledge discovery. Kluwer Academic Publishers, 1997.

[Hay99] Haykin, S., Neural Networks: A comprehensive foundation. Second Edition, Prentice Hall, 1999.

[Hec96] Heckerman, D., A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995.

[Jen98] Jensen, F.V., An introduction to Bayesian networks. London, England: UCL Press, 1998.

[Jor99] Jordan, M.I., Learning in graphical models. MIT Press, Cambridge, 1999.

[Lam94] Lam, W., and Bacchus, F. Learning Bayesian belief networks. An approach based on the MDL principle. Computational Intelligence, 10, 269–293.

[Mae00] Maes, S., Tuyls, K., and Vanschoenwinkel, B., Machine Learning Techniques for Fraud Detection. Master thesis, VUB, 2000.

[Pea88] Pearl, J., Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Mateo, CA, 1988.

[Ree99] Reed, R.D. and Marks, R.J., Neural Smithing: supervised learning in feed-forward artificial neural networks. MIT Press, 1999.

[Rus95] Russell, S., Norvig, P., Artificial intelligence: A modern approach. Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey, 1995.

A Review On Credit Card Fraud Detection Using Machine Learning

Suresh K Shirgave, Chetan J. Awati, Rashmi More, Sonam S. Patil

Abstract: In recent years credit card fraud has become one of the growing problem. A large financial loss has greatly affected individual person using credit card and also the merchants and banks. Machine learning is considered as one of the most successful technique to identify the fraud. This paper reviews different fraud detection techniques using machine learning and compare them using performance measure like accuracy, precision and specificity. The paper also proposes a FDS which uses supervised Random Forest algorithm. With this proposed system the accuracy of detecting fraud in credit card is increased. Further, the proposed system use learning to rank approach to rank the alert and also effectively addresses the problem concept drift in fraud detection.

Index Terms: Concept drift, credit card fraud, Machine Learning, Random Forest

1. INTRODUCTION

Credit card fraud is a major problem that involves payment card like credit card as illegal source of funds in transactions. Fraud is an illegal way to obtain goods and funds. The goal of such illegal transaction might be to get products without paying or gain an unauthorized fund from an account. Identifying such fraud is a troublesome and may risk the business and business organizations. In the real world FDS [1], investigator are not able to check all transactions. Here the Fraud Detection System monitors all the approved transactions and alerts the most suspicious one. Investigator verifies these alerts and provides FDS with feedback if the transaction was authorized or fraudulent. Verifying all the alerts everyday is a time consuming and costly process. Hence investigator is able to verify only few alerts each day. The rest of the transactions remain unchecked until customer identifies them and report them as a fraud. Also the techniques used for fraud and the cardholder spending behavior changes over time. This change in credit card transaction is called as concept drift [1] [7]. Hence most of the time it is difficult to identify the credit card fraud. Machine Learning is considered as one of the most successful technique for fraud identification. It uses classification and regression approach for recognizing fraud in credit card. The machine learning algorithms are divided into two types, supervised [14][18] and unsupervised [16] learning algorithm. Supervised learning algorithm uses labeled transactions for training the classifier whereas unsupervised learning algorithm uses peer group analysis [23] that groups customers according to their profile and identifies fraud based on customers spending behavior.

- *Dr. Suresh K. Shirgave is Associate Professor, DKTE Society's Textile and Engineering Institute, Ichalkaranji, India, E-mail: skshirgave@gmail.com*
- *Chetan J. Awati is Assistant Professor, Department of Technology, Shivaji University Kolhapur, India, E-mail: cja_tech@unishivaji.ac.in*
- *Rashmi More is M. Tech. Student, Department of Technology, Shivaji University Kolhapur, India, E-mail: rashmimore107@gmail.com*
- *Sonam S. Patil is Assistant Professor, Department of Information Technology, D Y Patil College of Engineering, Akurdi, Pune India, E-mail: skh9624@gmail.com*

Many learning algorithm have been presented for fraud detection in credit card which includes neural networks [14][19][21][22], Logistic Regression [3], decision tree [4][15], Naive Bayes [6], Support Vector Machines [5], K-Nearest Neighbors [6] and Random Forest [1][2]. This paper examines the performance of above algorithms based on their ability to classify whether the transaction was authorized or fraudulent and then compares them. The comparison is made using performance measure accuracy, specificity and precision. The result showed that Random Forest algorithm showed better accuracy and precision than other techniques.

2 RELATED WORK

There are different supervised and unsupervised learning algorithms used for fraud detection in credit card. Some important are described below. The author [1] has proposed a paper where they have first explained the proper performance measures which is used for fraud identification. The authors have structured a novel learning technique that can solve concept drift, verification latency, and class imbalance issues. The paper also showed effect of above issues in true credit card transactions. Here in paper [2] authors presented two types of classifier using random forests which are used to train the behavior features of transactions. The authors have compared the two random forests and have analyzed their performance on fraud identification in credit card. In paper [3] authors presented a FDS for credit card using Artificial Neural Network and Logistic Regression. The system used to monitor each transaction separately using classifier and then classifier would generate score for each transaction and label this transaction as legal or illegal transaction. A decision tree method was proposed in paper [4]. The method decreased overall misclassification costs and selected splitting property at each node. The author also compared the decision tree method for fraud identification with other models and proved that this approach performs well using performance measure like accuracy and genuine positive rate. The author [5] developed a FDS for credit card transaction using support vector machines and decision tree. This study built seven alternative models that were created using support vector machines and decision tree. The author also compared this classifiers performance using performance measure

accuracy. The study also showed that as size of training dataset increases the number of fraud detected by SVM are less than fraud identified by decision tree method. Here in [6] author presented fraud detection system using a Naive Bayes K-Nearest Neighbors method. The main aim of proposed system was to improve accuracy. Naive Bayes Classifier predicts probabilities of fraud in transaction while KNN classifier predicts how near the undefined sample data is to kth training dataset. The author compared both this classifier and showed that both work differently for given dataset. Most of predictive model used for detecting fraud in credit card transaction faces the issue of concept drift. The author [7] presented two FDS based on sliding window and ensemble learning and showed that classifier need to be trained separately using feedback and delayed samples. The outcome of the two was than aggregated to improve the alert precision in FDS. Thus the author showed that to solve the issue of concept drift, the feedback and delayed samples are to be handled separately.

3 COMPARISON

Performance of all learning algorithms used for fraud detection in credit card transactions are compared in table 1 The comparison is based on their accuracy, precision and specificity.

TABLE 1
COMPARISON OF MACHINE LEARNING TECHNIQUES

Classifiers	Metrics		
	Accuracy	Precision	Specificity
Random Forest	0.962	0.997	0.987
Logistic Regression	0.947	0.996	0.979
KNN	0.942	0.410	0.971
SVM	0.938	0.782	0.984
Decision Tree	0.908	0.91	0.912
Naive Bayes	0.937	0.505	0.9741

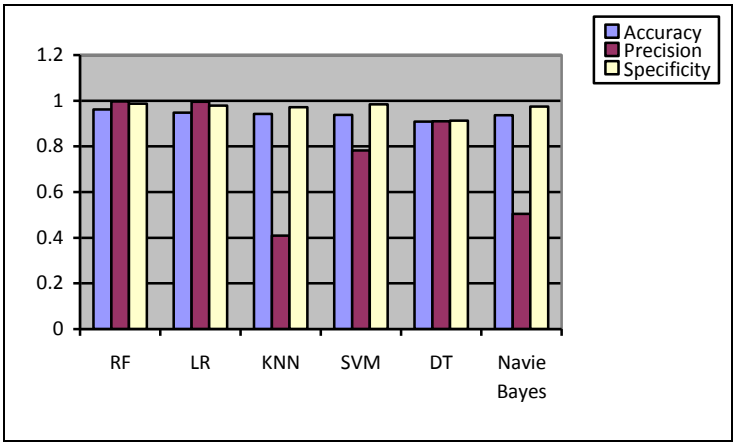


***Fig**. 1. Accuracy, precision and specificity performance of all classifier*

From table 1 we can see that accuracy of random forest is far better than the other learning algorithms. From fig. 1 we can see that precision, accuracy and specificity of Random Forest is highest followed by Logistic regression, SVM, Decision Tree, Naive Byes and KNN. Hence the proposed system using random forest will show better accuracy for larger number of training data.

4 PROPOSED SYSTEM

Today modern society is using credit cards for variety on reasons. Similarly fraud in credit card transactions has been growing in recent years. Each year, a huge amount of financial losses are caused by the illegal credit card transactions. Fraud may occur in variety of different forms and may be limited. Therefore there is need to solve the issues of fraud detection in credit card. Additionally, with the development of new technologies criminals finds new ways to commit fraud. To overcome this problem the proposed system for fraud detection in credit card transactions will be designed using ML technique that will provide investigator a small reliable fraud alerts.

4.1 Objectives

The proposed system will achieve following main objectives:

- To train the model using feedbacks and delayed samples and sum up their likelihood to identify alert.
- To implement machine learning technique to address concept drift and class imbalance issue.
- To develop a learning to rank approach to increase alert precision.
- To introduce performance measure those are considered in real-world FDS.

We propose a Fraud Detection System (FDS), which mainly focuses on data driven model and learning to rank method. It also focuses on alert feedback interaction that checks the way recent supervised samples are provided. Fig. 2 shows the block diagram of proposed system.

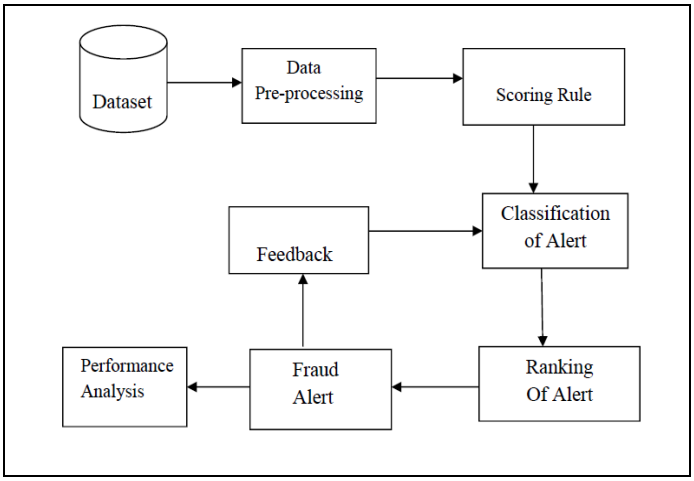


***Fig. 2**. Block Diagram of Proposed System*

4.2 Modules

The system is proposed to have the following modules along with functional requirements.

a. Data Preprocessing
b. Scoring Rule
c. Classification of Alerts
d. Ranking of Alert
e. Performance Analysis

4.2.1 Data Preprocessing

In this module selected data is formatted, cleaned and sampled. The data preprocessing steps includes following:

a. Formatting: The data which is been selected may not be in a suitable format. The data may be in a file format and we may like it in relational database or vice versa.
b. Cleaning: Removal or fixing of missing data is called as cleaning. The dataset may contain record which may be incomplete or it may have null values. Such records need to remove.
c. Sampling: As number of frauds in dataset is less than overall transaction, class distribution is unbalanced in credit card transaction. Hence sampling method is used to solve this issue.

4.2.2 Scoring Rule

Percentage of fraud in transaction is called as score. This module assigns score by matching recent transaction pattern with the past transaction pattern of cardholder. If score is greater then the transaction is considered as suspicious and further proceeding is stopped. Otherwise it is moved to next module.

4.2.3 Classification of Alert

Here machine learning model will be used that will train and update the data based on feedback and delayed samples. Classifier will be trained separately using feedback and delayed samples and their probabilities will be aggregated to identify alerts. Transaction that will be having high probability will be alerted. Hence only limited number of alerted transaction is reported to investigators.

4.2.4 Ranking of Alert

This module, rank each alert based on correctness of security question. This security questions will be created every time whenever the transaction is identified to be suspicious. The alerts are ranked using likelihood. If it is found that an alert has greater probability than other alerts then it is added to a queue and location of fraudster is tracked. This feature makes system user friendly and helps to file complaint against fraud.

5 CONCLUSION

This paper has reviewed various machine learning algorithm detect fraud in credit card transaction. The performances of all this techniques are examined based on accuracy, precision and specificity metrics. We have selected supervised learning technique Random Forest to classify the alert as fraudulent or authorized. This classifier will be trained using feedback and delayed supervised sample. Next it will aggregate each probability to detect alerts. Further we proposed learning to rank approach where alert will be ranked based on priority. The suggested method will be able to solve the class imbalance and concept drift problem. Future work will include applying semi-supervised learning methods for classification of alert in FDS.

REFERENCES

[1] Jalinus, N., Nabawi, R. A., & Mardin, A. (2017). The Seven Steps of Project-Based Learning Model to Enhance Productive Competences of Vocational Students. In 1st International Conference on Technology and Vocational Teacher (ICTVT 2017). Atlantis Press. Advances in Social Science, Education and Humanities research (Vol. 102, pp. 251-256).
[2] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi and Gianluca Botempi, "Credit card Fraud Detection : A realistic Modeling and a Novel Learning Strategy", IEEE Trans. on Neural Network and Learning system,vol.29,No.8, August 2018.
[3] Shiyang Xuan,Guanjun Liu,Zhenchuan Li,Lutao Zheng,Shuo Wang, Jiang,"Random Forest for credit card fraud detection",Int.conf.on Networking,Sensing and control,2018.
[4] Y. Sahin , and Duman,E.,(2011) "Detecting credit card fraud by ANN and logistic regression." In Innovations in Intelligent Systems and Applications(INISTA),2011 international Symposium on (pp.315-319).IEEE
[5] Y. Sahin, S. Bulkan, and E. Duman, "A cost-sensitive decision tree approach for fraud detection," Expert Syst. Appl., vol. 40, no. 15,pp. 5916–5923, 2013
[6] Sahin Y. and Duman E. (2011),"Detecting Credit Crad Fraud by Decision Trees and Support Vector Machines", International Multi-Conference Of Engineers and Computer Scientists(IMECS 2011),Mar 16-18,Hong Kong,Vol.1,pp.1-6
[7] Sai Kiran, Jypti Guru, Rishabh Kumar, Naveen Kumar, Deepak Katariya,"Credit card fraud detection using

Naïve Bayes model based and KNN classifier", Int. Journal of Adv. Research , Ideas and Innovations in Technology,vol.4,2018.

[8] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection and concept-drift adaptation with delayed supervised information," in Proc. Int. Joint Conf. Neural Netw., 2015,pp. 1–8.

[9] A. C. Bahnsen, D. Aouada, and B. Ottersten, "Example-dependent cost-sensitive decision trees," Expert Syst. Appl., vol. 42, no. 19,pp. 6609–6619, 2015

[10] A. Dal Pozzolo, O. Caelen, and G. Bontempi, "When is undersampling effective in unbalanced classification tasks?" in Machine Learning and Knowledge Discovery in Databases. Cambridge, U.K.: Springer, 2015

[11] N. Mahmoudi and E. Duman, "Detecting credit card fraud by modified fisher discriminant analysis," Expert Syst. Appl., vol. 42, no. 5,pp. 2510–2516, 2015

[12] A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, "Detecting credit card fraud using periodic features," in Proc. 14th Int. Conf. Mach. Learn. Appl., Dec. 2015, pp. 208–213.

[13] Anuruddha Thennakoon, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga,"Realtime Credit Card Fraud Detection Using Machine Learning ,"Int. Conf. on Cloud Computing, Data Science & Engineering,2019.

[14] S. Wang, L.L. Minku, and X. Yao, "Resampling-based ensemble methods for online class imbalance learning." Trans, Knowl., Data Eng., vol 27, no. 5, pp. 1356-1368, May 2015.

[15] Jan may Kumar Behera, Suvasini Panigrahi, "Credit Card Fraud Detection: A Hybrid Approach using Fuzzy Clustering and Neural Network", 2015 IEEE Second International Conference on Advances in Computing and Communication Engineering.

[16] M. Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, BankSealer: "A Decision Support System for Online Banking Fraud Analysis and Investigation", Berlin, Germany: Springer, 2014, pp. 380–394

[17] R. J. Bolton and D. J. Hand, "Unsupervised profiling methods for fraud detection," in Credit Scoring Credit Control VII. London, U.K.: Imperial College London, 2001, pp. 235–255

[18] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," Trans. Neural Netw., vol. 22, no. 10,pp. 1517–1531, 2011.

[19] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision SupportSyst., vol. 50, no. 3, pp. 602–613, 2011.

[20] Tao Guo ,Gui-Yang Li ,"Neural data mining for credit card fraud detection",Int.Conf. on Machine Learning and Cybernetics, Sept 2008

[21] J. Gao, B. Ding, W. Fan, J. Han, P.S. Yu, "Classifying data streams with skewed class distributions and concept drifts", IEEE internet comput., vol.12, no. 6, pp. 37-49, Nov 2008

[22] E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A neural network based database mining system for credit card fraud detection," in Proc. IEEE/IAFE Computat. Intell. Financial Eng., Mar. 1997, pp. 220–226.

[23] J.R. Dorronsoro, F. Ginel, C. Sgnchez and C.S. Cruz, "Neural fraud detection in credit card operations", IEEE transaction neural network vol. 8, no. 4, pp. 827-834, Jul.1997.

[24] D. J. Weston, D. J. Hand, N. M. Adams, C. Whitrow, and P. Juszczak,"Plastic card fraud detection using peer group analysis," Adv. Data Anal.Classification, vol. 2, no. 1, pp. 45–62, 2008.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-9, Issue-3, May 2021
https://doi.org/10.21276/ijircst.2021.9.3.5
Article ID IRP1151, Pages 31-36
www.ijircst.org

Analysing Auto ML Model for Credit Card Fraud Detection

Vaishali Garg[1], Sarika Chaudhary[2] and Anil Mishra[3]

[1] Student, Department of Computer Science & Engineering, Amity University, Gurugram, India
[2] Assistant Professor, Department of Computer Science & Engineering, Amity University, Gurugram, India
[3] Assistant Professor, Department of Computer Science & Engineering, Amity University, Gurugram, India

Correspondence should be addressed to Vaishali Garg; vishugarg290@gmail.com

ABSTRACT- Fraud Detection is a major concern these days because of digitalization. We are totally dependent on online transactions these days for even very small needs. There is no doubt that online transactions have made our life very easy but it has increased risk on other hand. And this risk can be very harmful one day. Confidential data is being stolen by the different apps and it is sold in international market. Which later on comes to us in totally different and very harmful way. So why not to use technology again to stop these risks and flaws. Various ML techniques has been observed by researchers but Auto ML is yet not discovered on a wider platform. Therefore, this paper at first aims to explore the trending technology Auto ML. Then a model for evaluating Auto ML is suggested and analysed with different classification algorithms. The experimental results ascertained the accuracy of Auto ML followed by a comparative analysis of ML and Auto ML.

KEYWORDS- Auto ML, Classification, Credit card, Fraud detection, Machine learning

I. INTRODUCTION

For years, fraud has been a serious issue in sectors like banking, medical, insurance, and lots of others. Due to the rise in online transactions through different payment options, like credit/debit cards, PhonePe, Gpay, Paytm, etc., fraudulent activities have also increased. Moreover, fraudsters or criminals became very skilled find escapes in order that they will loot more. Since no system is perfect and there is always a loophole them, it has become a challenging task to make a secure system for authentication and preventing customers from fraud [1]. So, Fraud detection algorithms are very useful for preventing frauds. The rapid growth in E-Commerce industry has led to an exponential increase in the use of credit cards for online purchases and consequently they has been surge in the fraud related to it. Machine learning plays an important role for detecting the master-card fraud within the transactions. Credit card fraud is the most common form of identity theft, affecting more than 10.7 million people annually [2]. It occurs when someone steals a card or snatches personal information to perform so-called card-not-present (CNP) transactions.

Automated machine learning provides methods and processes that enable machine learning professionals to access machine learning without machine learning, in order to improve machine learning efficiency and accelerate machine learning research. In recent years, machine learning (ML) has made great progress, and more and more disciplines are relying on it[3]. However, the key to this achievement rest on the grade to which machine learning experts accomplish the following tasks:

- Pre-process and clean the data.
- Select and construct appropriate features.
- Select an appropriate model family.
- Optimize model hyper-parameters.
- Post-process machine learning models.
- Critically analyse the results obtained.

As the difficulty of these tasks is a lot beyond non-ML-experts, the quick evolution of machine learning applications has formed a demand for off-the-shelf machine learning methods that can be used easily and without expert knowledge.

II. LITERATURE REVIEW

Fraud act because the unlawful or criminal deception intended to end in financial or personal benefit. It's a deliberate act that's against the law, rule or policy with an aim to achieve unauthorized financial benefit [4]. Numerous literatures concerning anomaly or fraud detection during this domain are published already and are available for public usage. A comprehensive survey conducted by Guedlek et al.[8] and his associates have revealed that techniques employed during this domain include data processing applications, automated fraud detection, adversarial detection. Albeit these methods and algorithms fetched an unexpected success in some areas, they did not provide a permanent and consistent solution to fraud detection. An identical research domain was presented by Quah et al.[6] where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transaction in an emulation experiment of MasterCard transaction data set of 1 certain full service bank . Outlier mining may be a field of knowledge mining which is essentially utilized in monetary and internet fields. It deals with detecting objects that are detached from the most system i.e. the transactions that aren't genuine [7][16]. They need taken attributes of customer's behaviour and supported the worth of these attributes they've calculated that distance between the observed value of that attribute and its predetermined value. There have also been

efforts to progress from a totally new aspect. Attempts are made to enhance the alert-feedback interaction just in case of fraudulent transactions [9]. Just in case of fraudulent transaction, the authorised system would be alerted and a feedback would be sent to deny the continued transaction. Table I shows the comparison among the various existing techniques for the detection of frauds [10]. Advantages and drawbacks of the techniques are discussed below.

Table 1: Summarized Fraud Detection Techniques

Fraud Detection Techniques	Observations	Limitations
K-nearest Neighbour Algorithm	Define anomalies in the target instance and is easy to implement.	Appropriate for detecting frauds with the limitations of memory.
Hidden Markov Model (HMM)[5]	Identify the fraudulent activity during transaction.	Unable to detect fraud with a less transactions.
Neural Network	Detect real-time credit card frauds.	Have many sub-techniques. So, if they pick-up this which is not suitable for credit card fraud detection, the performance of the method will decline.
Decision Tree	Handle non-linear credit card transaction as well.	DT cannot detect fraud at the real time of transaction.
Outlier Detection Method	Lesser memory and computation requirements. Works fast and well for large online datasets.	Cannot find anomalies accurately like other methods.
Deep Learning[14]	It can extract complex patterns	Only used in image recognition. No information to explain the other domains is available. The library of deep learning does not cover all algorithms.

After analysing the literature following gaps are identified:

- The major issue which comes into play is growing technology and with growing technology every day comes a new method of fraud especially in online transaction. Many companies do not reveal these frauds so as to protect their reputation, so most of the frauds remain unreported which leads to another harmful frauds.
- Another important issue is the maintenance of huge amount of database.
- It is very difficult to handle such a huge amount of data. Pre-processing of data takes so much time.
- Real time working problems as the incoming transactions are excessive and behavior of card holders and fraudsters change in rapid way.

III. PROPOSED MODEL DESIGN

This section describe the proposed model based upon the gaps identified. To detect the credit card fraud detection there are ample models that are available. But then the question arises which model to choose, which the best model is? There are many types of Machine Learning models specific to different use cases. As we work with datasets, a machine learning algorithm works in two stages. We usually split the data around 20%-80% between testing and training stages. Under supervised learning, we split a dataset into a training data and test data in Python ML. Fig 1. Depicts the workflow of the model.

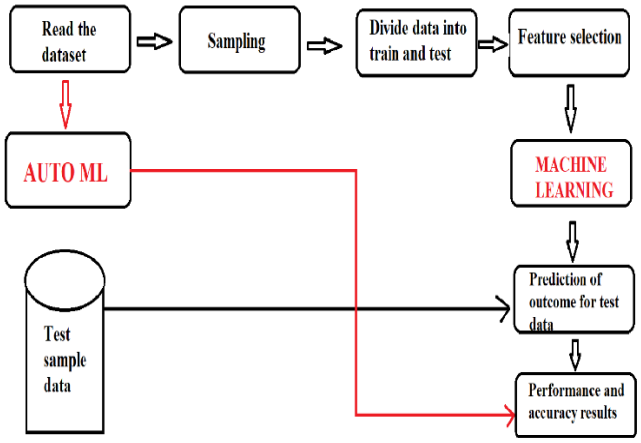


Fig. 1: Workflow of proposed model

A. Pre-processing

First of all the data is read using the panda's library. Data Pre-processing is that step in which the data gets transformed, or Encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

B. Oversampling

After the pre-processing step then comes the oversampling. As the data is highly imbalanced. We need to do oversampling to bring the data to the balanced state.

C. Splitting the dataset into test and train data

Entire dataset is divided into two parts. The train data set and test data set. 80% of the data is feed into training and rest 20% is feed into testing.

D. Feature Selection

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model

construction. After the feature selection the results are analyzed and accuracy is measured.

E. AUTO ML

This is the main part of the entire model. With few commands only auto ml compares different models and extra trees classifier model is being built for further prediction.

IV. RESULTS AND DISCUSSIONS

A. Dataset

The Data-set used in this work as depicted in fig. 2 contains the transactions made in two days by European cards in September 2012, gathered and analyzed during a research collaboration of Worldline and the Machine Learning Group of ULB on big data mining and fraud detection. It is freely available on Kaggle. The data contains only numerical values. Due to confidentiality the values where changed by PCA transformation. The features time and amount have not been transformed and all other features are represented by V0, V1……V26 values.

Table 2: Dataset Description

Variable name	Description	Type
V0,V1------ V26	Transaction features after transformation	Integer
Time	Time elapsed between each and the first transaction	Integer
Amount	Amount of transaction	Integer
Class	Non fraud or Fraud	0 or 1

B. Performance Metrics

This Data-set classifies transactions by being fraudulent or not. We have 492 frauds out of 284807, which is highly unbalanced 0.173%. To solve this class unbalance, Random over-Sampling is used. Over Sampling shows the distribution of the Data-set. After Over-Sampling dataset is spliced into training and test sets. For a Pre-trained model performance check, we split the data into two separate training sets and one independent test set for final model comparison. Table III shows the instances.

Table 3: Instances of the dataset

Number of instances	284807
Split ratio for pre – training	0.2
Split ratio for training	0.4
Independent test set	0.4

C. Evaluation Metrics

The evaluation of the model was carried out using the various evaluation metrics such as Accuracy, Precision, F1-score, Recall.

Accuracy: is defined as the number of correct predictions made by the model. It is the proportion of the total number of correct predictions [11].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision: defines the results classified as positive by the model, how many were actually positive. It is the number of items correctly identified as positive out of total true positives [12].

$$\text{Precision} = \frac{\text{true positive}}{\text{true positive} + \text{false positives}}$$

Recall: It is the number of items correctly identified as positive out of the total items classified as positive[13][15].

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false negatives}}$$

F1-Score: is the weighted average of the precision and the recall, it takes both false negatives and positives into the account and gives a better outlook especially in an uneven class distribution it is given as:

$$\text{F1 Score} = 2\left(\frac{\text{Precision} * \text{recall}}{\text{Precision} + \text{recall}}\right)$$

Where True positive (TP) represents data detected as fraudulent, True negative (TN) represents data detected as legitimate, False positive (FP) represents normal data detected as fraudulent, and False Negative (FN) is denoted as fraud data detected as normal[13].

D. Experimental results

The described model is evaluated using thirteen algorithms as described in the figure below. Python and Jupyter Notebook is utilized for implementation. Fig.2 shows the comparison of various existing models using auto ml on different parameters like accuracy, f-1 score, recall, time and precision.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-9, Issue-3, May 2021
https://doi.org/10.21276/ijircst.2021.9.3.5
Article ID IRP1151, Pages 31-36
www.ijircst.org

```
best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9996	0.9456	0.7959	0.9460	0.8635	0.8633	0.8670	17.059
rf	Random Forest Classifier	0.9995	0.9474	0.7807	0.9406	0.8528	0.8525	0.8565	125.848
lda	Linear Discriminant Analysis	0.9993	0.9009	0.7368	0.8514	0.7875	0.7871	0.7904	1.081
ada	Ada Boost Classifier	0.9992	0.9701	0.7003	0.8235	0.7556	0.7552	0.7583	39.809
lr	Logistic Regression	0.9991	0.9459	0.6045	0.8139	0.6900	0.6896	0.6991	7.421
dt	Decision Tree Classifier	0.9991	0.8763	0.7530	0.7449	0.7461	0.7456	0.7471	10.843
ridge	Ridge Classifier	0.9989	0.0000	0.4257	0.8214	0.5525	0.5520	0.5859	0.177
gbc	Gradient Boosting Classifier	0.9989	0.5691	0.4188	0.7796	0.5054	0.5050	0.5441	221.055
knn	K Neighbors Classifier	0.9984	0.6034	0.0586	0.8167	0.1083	0.1081	0.2137	2.501
svm	SVM - Linear Kernel	0.9982	0.0000	0.0000	0.0000	0.0000	-0.0001	-0.0002	5.727
lightgbm	Light Gradient Boosting Machine	0.9951	0.6923	0.5381	0.2131	0.2999	0.2982	0.3328	3.414
nb	Naive Bayes	0.9926	0.9662	0.6259	0.1370	0.2246	0.2224	0.2903	0.167
qda	Quadratic Discriminant Analysis	0.9758	0.9678	0.8667	0.0584	0.1093	0.1065	0.2212	0.597

Fig. 2: Comparison of various ML algorithms using auto ML

As analyzed from the above figure, extra tree classifier comes out to be the best algorithm with Auto ML. Fig.3 shows the Precision-Recall curve for extra trees classifier. Average precision comes out to be 0.73 and fig. 4 illustrate confusion matrix for the same.

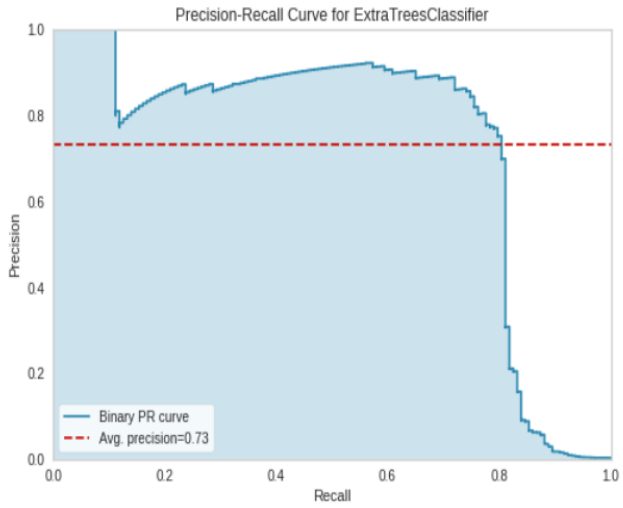


Fig.3: Precision-Recall curve for extra tree classifier

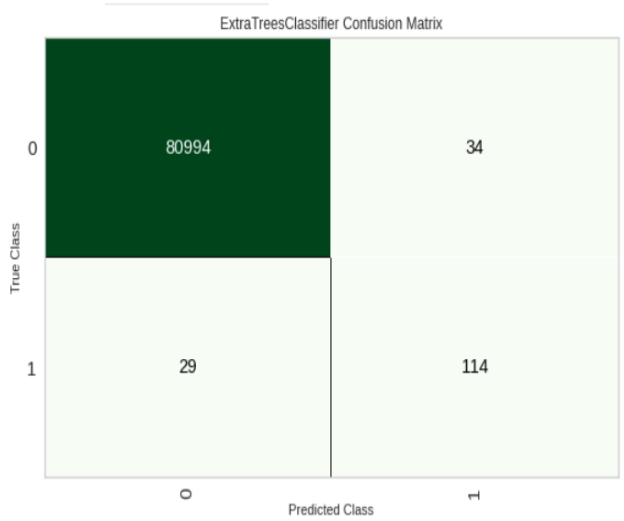


Fig.4: Confusion matrix of extra trees classifier

Fig.5 (a-d) shows the comparison of existing models using Auto ML on the basis of accuracy.

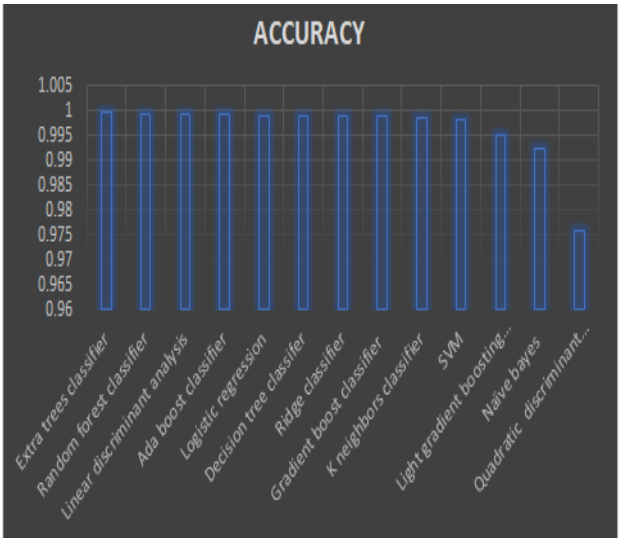


Fig. 5(a): Graph of existing models on the basis of accuracy

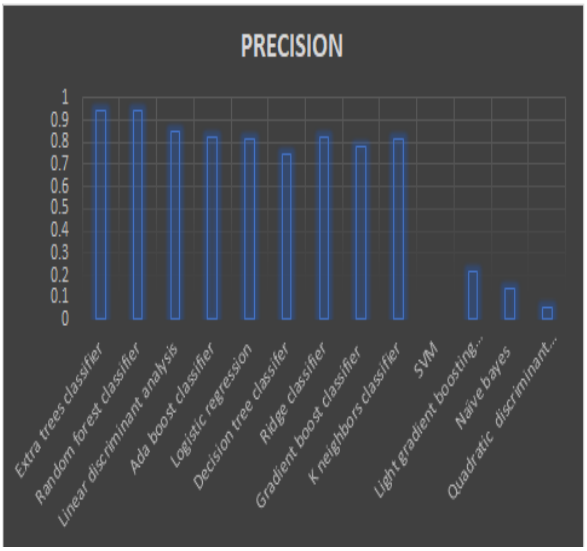


Fig. 5(d): Comparison of existing models on precision

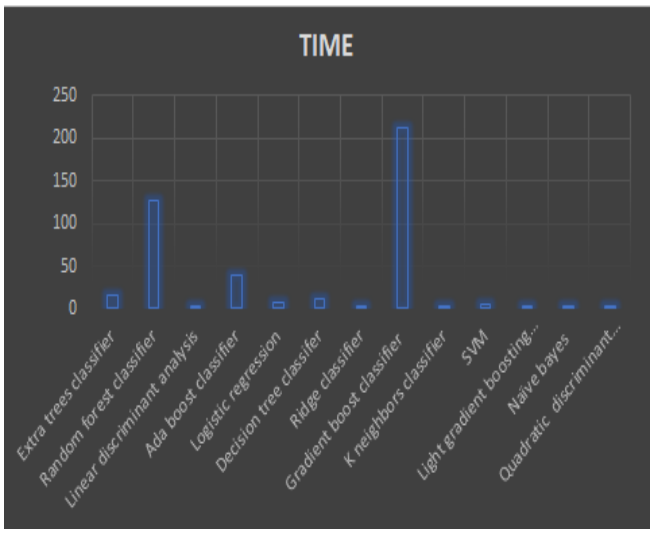


Fig.5(b): Graph of existing models on the basis of time.

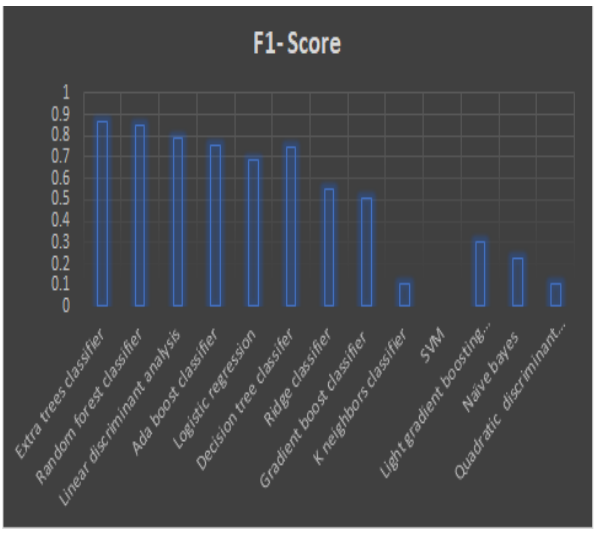


Fig. 5(e): Comparison of existing models on F1- score

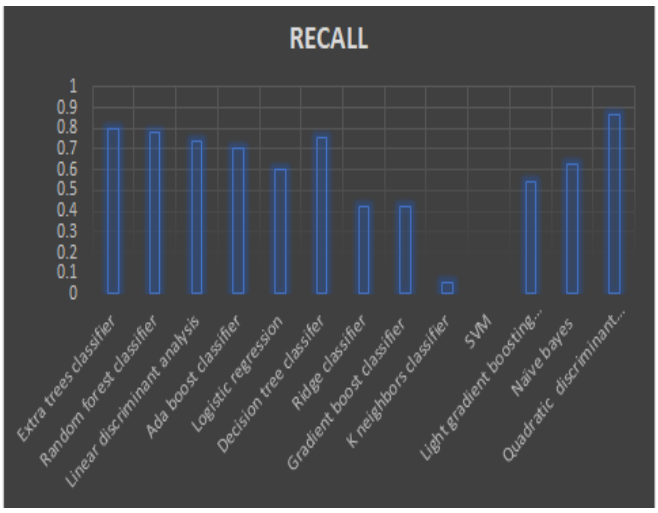


Fig. 5(c): Comparison of existing models on recall

Table 4: Comparative Analysis of Auto Ml and ML based on Experimental Evaluation

Feature	Auto ML	ML
Programming Complexity	Line of code is less	Line of code is more
Skill Requirement	Less skilled data scientists can also work	Skilled data scientists are needed to build models
Processing Time	Less	More
Hyper-Parameter Optimization	Present	Absent

V. CONCLUSION

In this paper different ML and Auto ML techniques are reviewed to discover the research gap. One of the significant observed approach in ML is classification. Mostly ML techniques have been utilized for credit card

fraud detection in past. Auto ML has still not discovered yet on a bigger platform for the same. Also, a novel auto ML based model is proposed. The model is capable to detect credit card fraud in comparison to various existing ML model in terms of processing time and easiness. Also, the quality of model is measured in terms of factors like accuracy, time, precision, recall. Then a comparative analysis of auto ml on existing models is being done and extra trees classifier comes out to be the best model on the factors like accuracy and time. The datasets examined for the analysis have been retrieved from online libraries, in future they can be directly collected from software industries to draw a fair and reasonable comparisons to measure the effectiveness of evaluation process .

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Maniraj, S., Saini, A., Ahmed, S., & Sarkar, S., "Credit card fraud detection using machine learning and data science". International Journal of Engineering Research and, 8(09) 2019.

[2] Patil, S., Nemade, V., & Soni, P. K. (2018). Predictive modelling for credit card fraud detection using data analytics. Procedia computer science, 132, 385-395.

[3] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019, March). Credit card fraud detection-machine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE.

[4] Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2019). Combining unsupervised and supervised learning in credit card fraud detection. Information Sciences.

[5] Srivastava, A., Kundu, A., Sural, S., & Majumdar, A. (2008). Credit card fraud detection using hidden Markov model. IEEE Transactions on dependable and secure computing, 5(1), 37-48.

[6] Quah, J. T., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. Expert systems with applications, 35(4), 1721-1732.

[7] S. Akila and U. Srinivasulu Reddy, "Cost-sensitive Risk Induced Bayesian Inference Bagging (RIBIB) for credit card fraud detection," Journal of Computational Science, vol. 27, pp. 247–254, Jul. 2018, doi: 10.1016/j.jocs.2018.06.009.

[8] M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications : A survey," Applied Soft Computing, vol. 93, p. 106384, Aug. 2020, doi: 10.1016/j.asoc.2020.106384.

[9] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: A comparative study," Decision Support Systems, vol. 50, no. 3, pp. 602–613, Feb. 2011, doi: 10.1016/j.dss.2010.08.008.

[10] G. C. de Sá, A. C. M. Pereira, and G. L. Pappa, "A customized classification algorithm for credit card fraud detection," Engineering Applications of Artificial Intelligence, vol. 72, pp. 21–29, Jun. 2018, doi: 10.1016/j.engappai.2018.03.011.

[11] Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," Information Sciences, May 2019, doi: 10.1016/j.ins.2019.05.042.

[12] S. M. S. Askari and M. A. Hussain, "IFDTC4.5: Intuitionistic fuzzy logic based decision tree for Etransactional fraud detection," Journal of Information Security and Applications, vol. 52, p. 102469, Jun. 2020, doi: 10.1016/j.jisa.2020.102469.

[13] C. S. Throckmorton, V. Mohan, J. M. William and C. Leslie, "Financial fraud detection using vocal, linguistic and financial cues," 2018.

[14] Y. Pandey, "Credit card fraud detection using deep learning" Int. J. Adv. Res. Comput. Sci., vol. 8, no. 5, May–Jun. 2017.

[15] Malik, Sanjay Kumar, and Sarika Chaudhary. "Comparative study of decision tree algorithms for data analysis." International Journal of research in Computer Engineering and Electronic. Page1 2 (2013).

[16] Kaur, Sonamdeep, Sarika Chaudhary, and Neha Bishnoi. "A Survey: Clustering Algorithms in Data Mining." International Journal of Computer Applications 975 (2015): 8887.

[17] Mandiratta, Sonam, Pooja Batra Nagpal, and Sarika Chaudhary. "A Perlustration of Various Image Segmentation Techniques." International Journal of Computer Applications 139.12 (2016).

[18]Sarika Chaudhary, Yojna Arora, Neelam Yadav (2020). Optimization of Random Forest Algorithm for Breast Cancer Detection IJIRCST Vol-8 Issue-3 Page No-63-66.

[19]Chaudhary,S., Nagpal.P (2019). "Live location tracker" , Global Research and Development Journal for Engineering, | Volume 4, Issue 10

ABOUT THE AUTHORS

Ms. Vaishali Garg is currently pursuing M.C.A from Amity University Haryana. She has published 4 research papers. Her research interest is Machine learning and deep learning.

Ms. Sarika Chaudhary is currently designated as Assistant Professor in CSE, Amity University, Haryana. She has published more than 34 research papers and 02 books. She is member of 16 Professional/Technical Committees and Editorial Board Member/Reviewer of 20 reputed Journals.

Dr. Anil Mishra is currently designated as Assistant Professor in CSE, Amity University, Haryana. He has published more than 16 research papers. He is an active member of ISTE, IEEE and CSI.

Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.92 – 95

RESEARCH ARTICLE

Credit Card Fraud Detection Using Decision Tree Induction Algorithm

[1]Snehal Patil, [2]Harshada Somavanshi, [3]Jyoti Gaikwad, [4]Amruta Deshmane, [#]Rinku Badgujar

Department Of Computer Engineering, JSPM's Bhivrabai Sawant Institute of Technology and Research, Wagholi Pune

[1] svpatil008@gmail.com, [2] harshada2910@gmail.com, [3] jrgaikwad121@gmail.com,

[4] deshmaneamruta55@gmail.com, [#] rinku.badgujar@gmail.com

Abstract-- Today Fraud is increasing all over the world, resulting in the vast financial losses. Chip and pin are developed for credit card systems, through fraud prevention mechanisms and these mechanisms do not prevent the most common fraud types such as fraudulent credit card usages over virtual POS (Point Of Sale) terminals or mail orders so known as an online credit card fraud. So as a result, fraud detection becomes the important tool and possibly the best way to stop such types of fraud. A new cost-sensitive decision tree approach which reduces the sum of misclassification costs while selecting the splitting attribute at each non-terminal node is advanced and the act of this approach is compared with the well-known traditional classification models on a real world credit card data set. The data mining layers prevent fraudsters to attack and improve a safe transaction. This research is totally concerned with credit card application fraud detection by performing the process of asking security queries to the persons intricate with the transactions and as well as by eliminating real time data faults.

Keywords— E-Commerce Security, Credit Card Fraud Detection, Data Mining, ID3 Decision Tree, Visual Cryptography

I. INTRODUCTION

The use of credit cards has increased, because of a rapid advancement in the electronic commerce technology. The credit card becomes the most popular type of payment for both online as well as regular purchase, cases of credit card fraud also increasing. In Modern day the fraud is one of the major effects of great commercial losses, not only for merchants, the individual clients are also affected.

Credit Card Fraud: Credit card fraud has been distributed into two types: Offline fraud and On-line fraud.

- The Offline fraud is dedicated by using a stolen physical card at call center or any other place.
- The On-line fraud is dedicated via phone, shopping, web, internet or in absence of card holder.

In the commercial practice a large-scale data-mining techniques can improve on the state of the art. The scalable techniques to analyze massive amounts of transaction data that powerfully compute fraud detectors in a timely manner is an important problem, especially for e-commerce. Moreover scalability and efficiency, the fraud-detection job exhibits technical difficulties that include slanted distributions of training data and non-uniform cost per error, both of which have not been usually studied in the knowledge-discovery and data mining community.

The use of machine learning in fraud detection has been an exciting topic. Due to the confidentiality of financial information and non-availability of public databases, few researches have the chance to work on developing methods exact to credit card fraud detection. However, the literature on credit card fraud detection is increasing and it has been shown that machine learning can be used effectively for this problem, in particular: neural

networks, artificial immune systems, association rules, Bayesian learning, support vector machines, and peer group analysis.

II. IMPLEMENTATION

A. Decision Tree Induction algorithm

A Decision tree algorithms are a method for approaching discrete-valued target functions, in which the learned function is denoted by a decision tree. These types of algorithms are famous in inductive learning and have been successfully applied to abroad range of tasks. We examine the decision tree learning algorithm – ID3.
The decision tree is a structure that contains root node, branch and leaf node. Every internal node indicates a test on attribute, every branch indicates the outcome of test and each leaf node holds the class tag. The uppermost node in the tree is the root node .A Decision trees organize circumstances by sorting them down the tree from the root to some leaf node, which delivers the classification of the instance. Each node in the tree specifies a test of some attribute of the instance and each branch descending from that node links to one of the possible values for this attribute. For example figure below explains a decision tree based on attribute name outlook.

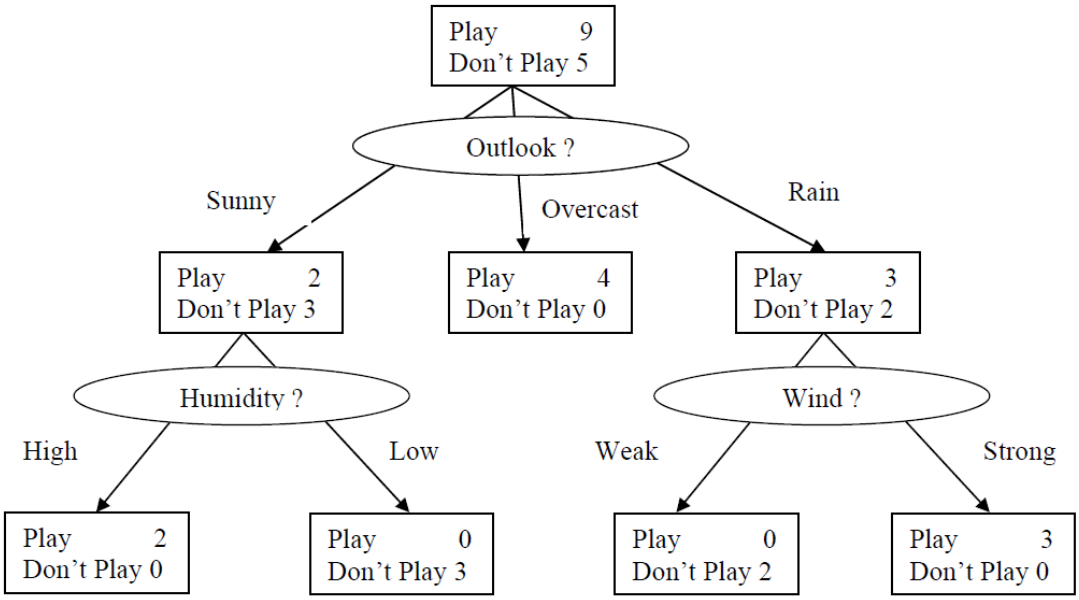


Fig. Decision Tree

1.1 Entropy

The entropy is a measure in the information theory, which illustrates the impurity of an arbitrary collection of examples. For e.g. if training data has 14 instances with 6 positive and 8 negative instances, the entropy is calculated as

Entropy ([6+, 8-]) = −(6 /14) log2 (6 /14) − (8/14) log2 (8/14) = 0.985

A key point to reminder here is that the more uniform is the probability distribution, the greater is its entropy.

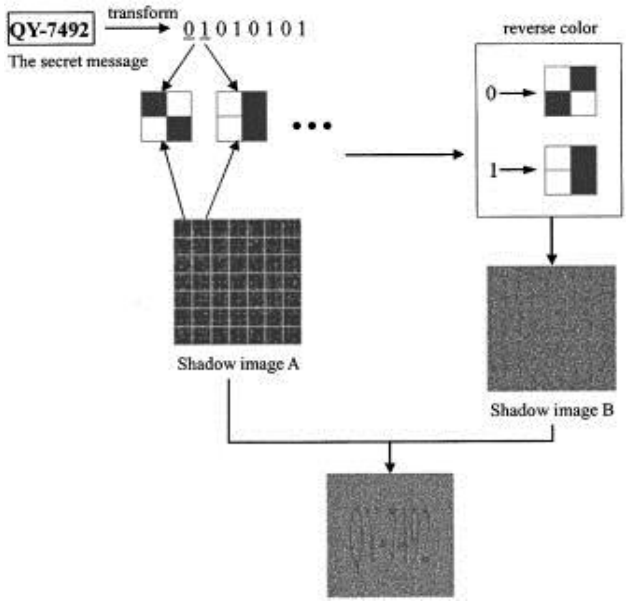
1.2 Information gain

The information gain measures the likely reduction in entropy by partitioning the examples according to the attribute.

B. Visual Cryptography

A visual cryptography is a kind of cryptography that can be decoded openly by the human visual system without any special calculation for decryption. The visual information text, allows through a cryptographic technique which is known as Visual cryptography to be encrypted in such a way that decryption becomes a mechanical operation that does not require a computer.
Visual cryptography is a popular solution for image encryption. The encryption procedure encrypts a secret image into the shares which are noise-like secure images which can be communicated or distributed over an untrusted communication channel by using secret sharing concepts. The secret image is decrypted without additional computations and any knowledge of cryptography by using the properties of the HVS to force the

recognition of a secret message from overlapping shares. The Visual cryptography is proposed in 1994 by Naor and Shamir who introduced a simple but perfectly secure way, which allows secret sharing without any cryptographic computation, which they known as Visual Cryptography Scheme (VCS). The simplest Visual Cryptography Scheme is given by the idea of a secret image contains of a collection of black and white pixels where each pixel is treated individually. There are many algorithms to encrypt the image in a different image, but a exceptional of them have been in visual cryptography for colour image. So we are using visual cryptography method.



Visual cryptography uses the characteristics of human vision which is an emerging cryptography technology, for decrypting the encrypted images. It wants neither complex computation nor cryptography knowledge. For security concerns it also ensures that hackers cannot perceive any clues about a secret image from individual cover images.

III.RESULTS

The procedure of decision tree for classification problems includes two steps: using a training data set to construct a decision tree; for each of the elements, applying decision tree to determine the elemental groups. Table shows an example includes the ten articles about fraud and related information, and records as data set S, now ID3 algorithm is used to create fraud classification decision tree about the credit card:

Number (Not as decision attribute)	Sex(Sex)	Marital Status(Marraige)	Level of Education(Education)	Place of crime(Place)	Is It Fraud
1	Man	Unmarried	High school and below	Bank	Yes
2	Women	Married	Junior	Supermarket	No
3	Women	Unmarried	Undergraduate	Bank	No
4	Man	Married	Undergraduate	Other	No
5	Man	Other	Graduate and above	other	No
6	Women	Unmarried	High school and below	Bank	Yes
7	Man	Married	Undergraduate	Supermarket	No
8	Man	Other	Junior	Bank	Yes
9	Man	Unmarried	High school and below	Other	No
10	Women	Married	Undergraduate	Supermarket	Yes

Table. Data Set Of Fraud Information

Every data has five properties in the table, which has four attributes need to calculate their information gain, and takes this as the source to build a decision tree, the four properties are:
Sex, Marital status, Level of education, Place of crime.

IV. CONCLUSION

As Credit card fraud has become more and more widespread in recent years. To increase merchants risk management level in an automatic and active way, structure an perfect and easy handling credit card risk monitoring system is one of the key tasks for the merchant banks. So we propose credit card fraud detection problem for the resolution of reducing the bank's risk. With the historical profile patterns, make use of credit card fraud detection models to equal the transaction information to predict the probability of being fraudulent for a new transaction. It offers a scientific basis for the authorization mechanisms.

V. ACKNOWLEDGEMENT

We acknowledge the effort and hard work by the experts who have contributed towards the development of Credit Card Fraud Detection System. We also acknowledge the reviewers of the journal for the suggestions and modifications to improve the quality of the paper.

REFERENCES

[1] Alejandro Correa Bahnsen, AleksandarStojanovic, DjamilaAouada and Bj¨ornOttersten" Cost Sensitive Credit Card Fraud Detection using Bayes Minimum Risk".
[2] Tatsuya Minegishi, AyahikoNiimi "Detection of Fraud Use of Credit Card by Extended VFDT".
[3] V.Dheepa,Dr. R.Dhanapal"Analysis of Credit Card Fraud Detection Methods". International Journal of RecentTrends in Engineering, Vol 2, No. 3, November 2009.
[4] Krishna Kumar Tripathi, Mahesh A. Pavaskar" Survey on Credit Card Fraud Detection Methods". International Journal of Emerging Technology and Advanced Engineering,Volume 2, Issue 11, November 2012.
[5] Y. Sahin and E. Duman" Detecting Credit Card Fraud by Decision Treesand Support Vector Machines".
[6] AlkaHerenj,Susmita Mishra" Secure Mechanism for Credit Card TransactionFraud Detection System". International Journal of Advanced Research in Computer and Communication EngineeringVol. 2, Issue 2, February 2013
[7]AnandBahety,Department of Computer Science" Extension and Evaluation of ID3 – Decision Tree Algorithm".
[8] KaiqiZou, Wenming Sun" ID3 decision tree in fraud detection application".
[9] SozanAbdulla"New Visual Cryptography Algorithm ForColored Image"Volume 2, Issue 4, April 2010.
[10]SnehalPatil,HarshadaSomavanshi,JyotiGaikwad,AmrutaDeshmane,Dept of Computer Engg JSPM'S BSIOTR Pune, India "Credit Card Fraud Detection Using Induction Algorithm".

ACEEE *Proc. of Int. Conf.* on *Recent Trends in Information, Telecommunication and Computing, ITC*

Real-Time Credit-Card Fraud Detection using Artificial Neural Network Tuned by Simulated Annealing Algorithm

Azeem Ush Shan Khan, Nadeem Akhtar and Mohammad Naved Qureshi
Aligarh Muslim University, Department of Computer Engineering, Aligarh, India
Email: azeem5257@gmail.com
Aligarh Muslim University, Department of Computer Engineering, Aligarh, India
Email: {nadeemalakhtar, navedmohd786}@gmail.com

***Abstract*— Now-a-days, Internet has become an important part of human's life, a person can shop, invest, and perform all the banking task online. Almost, all the organizations have their own website, where customer can perform all the task like shopping, they only have to provide their credit card details. Online banking and e-commerce organizations have been experiencing the increase in credit card transaction and other modes of on-line transaction. Due to this credit card fraud becomes a very popular issue for credit card industry, it causes many financial losses for customer and also for the organization. Many techniques like Decision Tree, Neural Networks, Genetic Algorithm based on modern techniques like Artificial Intelligence, Machine Learning, and Fuzzy Logic have been already developed for credit card fraud detection. In this paper, an evolutionary Simulated Annealing algorithm is used to train the Neural Networks for Credit Card fraud detection in real-time scenario. This paper shows how this technique can be used for credit card fraud detection and present all the detailed experimental results found when using this technique on real world financial data (data are taken from UCI repository) to show the effectiveness of this technique. The algorithm used in this paper are likely beneficial for the organizations and for individual users in terms of cost and time efficiency. Still there are many cases which are misclassified i.e. A genuine customer is classified as fraud customer or vise-versa.**

***Index Terms*— Credit Card fraud detection, Simulated Annealing, Machine Learning, Training, Classification, Artificial Neural Network (ANN), Activation Function.**

I. INTRODUCTION

Credit card fraud is a kind of theft or unauthorized activity to make payment using credit card in an electronic payment system as a fake source of fund. The purpose of credit card fraud is to obtain money or make payment without owner permission. It involves illegal use of card or card information without the owner permission though it is a criminal deception and banned by laws. Because of the advancement in technology and software's, users can hide their identity and locations while committing any transaction over the web, which increases the fraud over the web. There are many methods of credit card fraud depicted in "figure 1", all are mentioned in details "see [1]", are as follows:

1. First fraud is that user using its own credit card knowing that they have no money in his credit card and

DOI: 02.ITC.2014.5.65

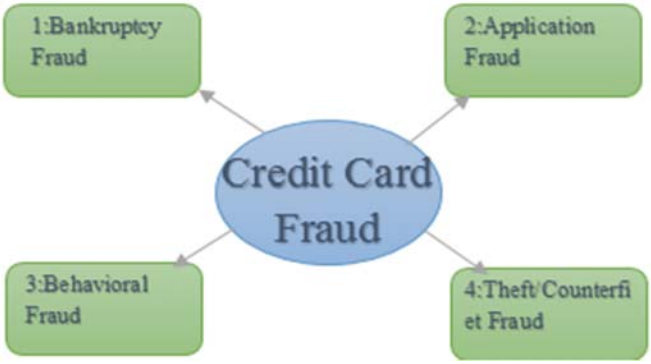


Fig 1: Types of Credit Card Fraud

bank has to pay by sending bill to the address.

2. Second fraud is done when submitting the application form to the bank for issuing the credit card with the fake information.
3. Third fraud is done online when purchasing any item by submitting the card information of any credit card without letting known to the owner.
4. Fourth fraud is the stealing of any credit card and using it by showing as an owner of this card until that card become blocked by bank.

Credit card fraud affect the organization by financial losses and individual user also affected if the information of credit card get steal. So it is important to find a solution which classifies a transaction into fraud or non-fraud. Many techniques have been developed for credit card fraud detection like Artificial Intelligence & Machine Learning and also based on locations [2]. In this paper, we focus on Machine Learning technique, basically it provides a system which is supposed to classify a current transaction into fraud or non-fraud.

In this paper, we are taking credit card fraud detection problem as a classification problem. Many classification algorithm have been developed [3], but the most popular one is Decision Tree. This algorithm is already been proposed for credit card fraud detection problem ("see [4]"). Basically there are two technique for credit card fraud detection:

1. Supervised
2. Un-supervised

These are the machine learning techniques, in which the first one uses training data, to build the model, which have all the attributes including class label i.e. it already contains the attribute which tells whether this previous transaction is fraud or not. And in the second technique, training data does not contain the class label i.e. this technique is class less. More study on these can be found in [5]. This paper propose a credit card fraud detection technique using Neural Networks and Simulated Annealing algorithm to adjust the weight of the neural network, Neural Network is a supervised machine learning technique. Many research papers has already been proposed for fraud detection using Neural Networks and many researchers have been uses Genetic Algorithm to adjust the weight of Neural Network in different fields [6]. This study of using Simulated Annealing to train Neural Network is one of the first to use for credit card fraud detection on real data set provided by UCI repository [7]. Basically, annealing is a process of heating and then cooling a solid to change the hardness of the solid and simulated annealing is to emulate this process.

The main aim of this process is to build a training model on the basis of previous transactions, called training data, for fraud detection. Once a learning of training model is complete, the model is capable of classifying the unseen online transaction as fraudulent or non-fraudulent in real time [8] & [9].

II. Artificial Neural Network

Artificial neural network works in the same way as a human brain does, human brain consist of number of neurons connected with each other, in the same way ANN consists of artificial neurons, called nodes in network, connected with each other. The idea of Artificial Neural Network was presented in late 1943 by Walter Pitts and Warren S.McCulloch as a data processing unit for classification or prediction problems [10]. For the first time, Dorronsoro "et al." in 1997 developed a system to detect credit card fraud by using

Neural Network. Now-a-days, ANN have been successfully applied in business failure prediction, stock price prediction, credit fraud detection and many more area.
ANN comes in many forms like Recurrent NN, Associative NN, etc. In this paper, we will discuss Feed-Forward Neural Network which will trained by Simulated Annealing method.
"Figure 2" depict a simple multi-layer feed forward neural network. It consist of an Input layer, an output layer and an Hidden Layer, hidden layer depends on the problem we are going to solve, it can be no or more than one hidden layer. The number of neurons in input layer corresponds to the number of input attributes in the training dataset which we will see later in this paper and the number of neurons in output layer is depend on the type of problem you are going to solve, in credit card fraud detection case we have two output one is fraud and the other one is non-fraud i.e., 0 and 1 respectively.
In the Feed-forward neural network, as we can see in "fig. 2", there is no feedback loop. Each of the neurons in each layer is connected with each other without making any loop and the link between these neurons has weights, represented by Wij. The connection between each neuron do not perform any calculation but is used to store the weights. These weights are initialized with some random values and changes at every iteration in training process. The simple neuron in each layer are often called perceptron is the simplest neuron network.

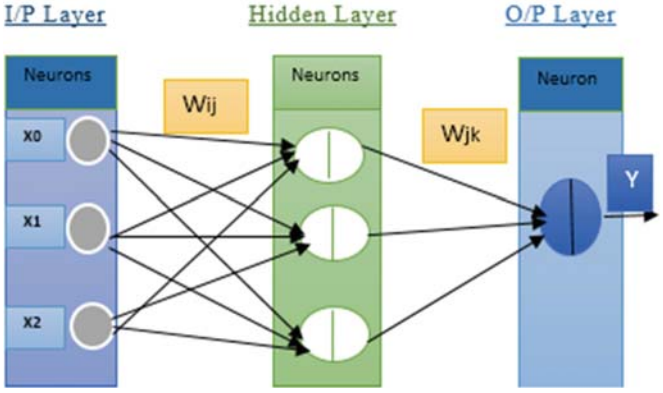


Fig 2: Simple Feed-Forward Neural Network

A feed-forward perceptron works by sending the input to the neurons and send to the output neuron after processing. This is a simple neuron, i.e. perceptron, "figure 3" which has three input and two output.

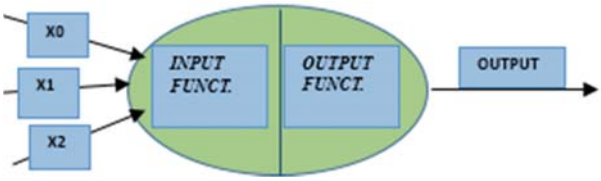


Fig 3: Simple Feed-Forward Perceptron

All the perceptron in the Neural Network have two functions i.e. Input and Activation Function. As the name suggest, the input function collects all the input and perform summation function on the input and then transfer the result to the activation function. An activation function perform some operation on the result after summation and then transfer to the next level. Let's take an example, in the above figure we have three inputs, let's say, I0, I1, I2 and two output Z1, Z2 and their corresponding weights. Now, input function will perform the summation on the inputs multiplied by their corresponding weights. Let's say the output of input function is S.

$$S = \sum_{2}^{x=0} Ix.Zx$$

The result of this summation function is then pass to activation function. Activation function scale the value of S in proper range. Common activation function are sigmoid activation function which works on threshold, if the value of S exceeded the threshold value then the node pass output.

There are two activation function which is commonly used in Neural Networks, Sigmoid and Hyperbolic Tangent Activation Function. It depends on the training dataset on which we are going to train the network that which activation function is good.
The "figure 4" shows sigmoid activation function graph, which refers to one of the case of logistic function. It works for real input values and it only returns positive value ("refer [11]"). The formula of sigmoid function is:

$$S(t) = \frac{1}{1 + e^{-t}}$$

The Hyperbolic Tangent activation function (TANH) is the next version of sigmoid function because it produces both negative as well as positive values as shown in "figure 5".

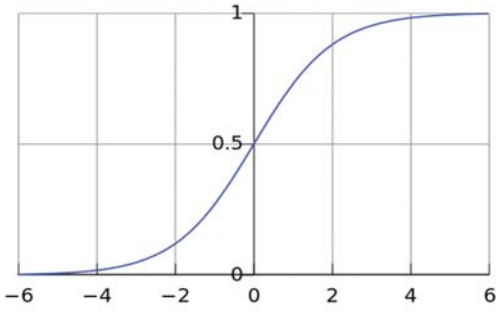


Fig 4: Sigmoid Activation Function Graph

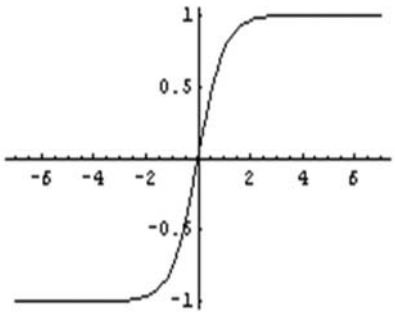


Fig 5: Hyperbolic Tangent Activation Function Graph

The equation of hyperbolic tangent function is given by:

$$F(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

In this project, we have use both of the above activation function and the result of fraud detection is better with TANH.

III. Simulated Annealing

Basically Annealing is a thermodynamics process, it is a heat treatment process upon any metal to change the structure of the metal. It involves heating of any metal slightly above its critical temperature and then cooling it down slowly. It makes the metal harder or stronger and makes the structure of the metal homogenous. The emulation of the process of annealing is called Simulated Annealing.
This method was developed by adapting some changes in Metropolis-Hastings algorithm, also known as Monte Carlo method, invented by M.N. Rosenbluth and published in a paper in 1953 [12]. This method was developed by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983 [13], and later on by Vlado Cerny in 1985 [14]. Corana "et al." (1987) and Goffe (1994) had proposed some changes which was suitable to train discrete-valued weights. In this study, the implementation of simulated annealing is based on these algorithms, which is adjusted to find the best configuration of weights in artificial neural network. The basic procedure are as follows:

1. Heat the system at high temperature T and generate a random solution.
2. As the algorithm progress, T decreases at each iteration and each iteration forms a nearby model.
3. Then cool the system slowly until the minimum value of T is reached and generate a model at each iteration, which takes the system towards global minima.

In each iteration, a solution is generated which is compared with current solution, by using acceptance function, if it is better than current solution than it get replaced by this solution. The terminology and definitions used in Simulated Annealing is defined in [15] and [16]. These definitions are used in this paper to train the neural network for fraud detection. The main definitions which is needed for this algorithm are: (1) a method is to generate initial solution, by generating worst solution at the beginning helps to avoid converging to local minimum, (2) a Perturbation Function to find a next solution with whom the current solution is compared, (3) an Objective Function is to be defined to evaluate and rate the current solution on

the basis of performance, (4) an Acceptance Function, which is used to check whether the current solution is good or not in comparison with the current one, a very basic one is exp((currentSol-nextSol)/currentTemp), (5) and the last one is stopping criteria, there are many stopping criteria's, in this paper we have used an threshold value of objective function as an stopping criteria.

IV. Training Of Ann

ANN is made up of connection between neurons in each layer and links connecting these neurons has some weights on it, so the adjustment of weights to learn the relationship between the input and the given output i.e. label, is called learning or training of neural network. The most popular training algorithm is BackPropogation which was given by Salchenberger "et al." in 1992. The main problem of this algorithm is that it gets stuck in local minima and the error still remains the same. An evolutionary algorithm, Simulated Annealing and Genetic Algorithm, was given to solve this problem of local minima, among these algorithm simulated annealing is preferred because it takes less time in comparison with genetic algorithm.
As we know, to perform training of ANN we should have millions of data but to train neural network for credit card fraud detection we don't have much amount of data of previous transaction to perform training upon. In this paper, we have used Simulated Annealing for training and it gives very good result in comparison with genetic and backpropogation, which we will see later.

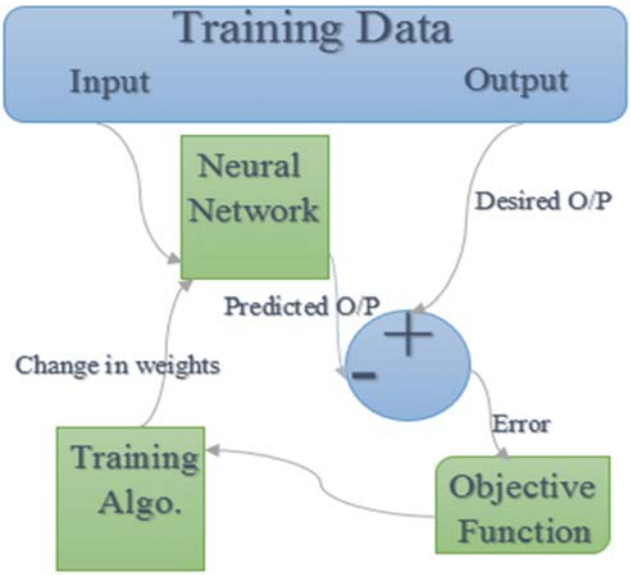


Fig 6: ANN training model

"Figure 6" depicts a basic model for the training process of ANN. In this paper we have used supervised learning [17], so our data consist of both input and desired output. A random weight is generated for each connection and output is calculated based on current weight & input. Obviously, in the initial states the desired output is different from current output which can be calculate by using any error function like Mean Squared Error (MSE) or Sum Squared Error (SSE). Now, according to the training algorithm the weights are adjusted and repeat these steps again until some threshold value for error function will reached.
In 1988, Jonathan Engel publish a paper [18] in which he had explained the training process for feed-forward NN using simulated annealing, we have used this paper to implement the simulated annealing algorithm for training purpose, while a brief description of algorithm is given below in this paper(for pseudo code refer [19]). There are the series of different steps which simulated annealing algorithm has to follow at each cycle. A cycle is completed when it follows all the steps shown in the "fig. 7" and randomized the weights at each cycle. The 'n' number of cycle is fixed by the programmer, at each iteration it will perform n cycles and after one iteration is completed, the current temperature gets lowered and checked against the minimum allowed lowest temperature, if it is not less than the threshold value then again a cycle of randomization is repeated. The method used in this paper for temperature reduction is based on start and stop temperatures. Its equation is given by:

NewTemp= Ratio * currentTemp

The ratio causes the new temperature lies in between start and stop temperature, ratio is calculated at each cycle and it decreases the temperature logarithmically. Its equation is given by:

$$Ratio = \frac{e^{log(stopTemp/startTemp))}}{cycles - 1}$$

The values of start and stop temperature is decided by hit and trial method, you have to check the result by putting different values and compare to find the best one. The high temperature will cause more randomization in weights.

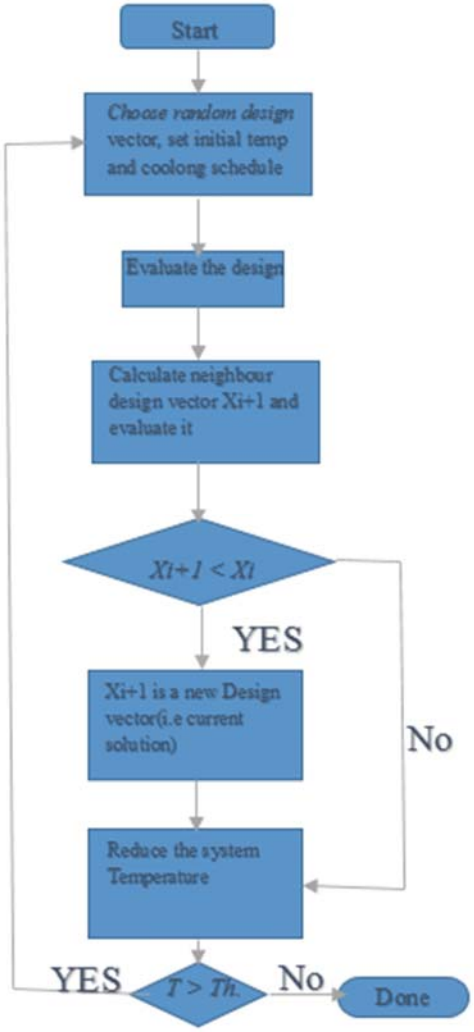


Fig 7: Simulated Annealing method for training ANN

The main part in the training process of an ANN is the randomization of weights, simulated annealing uses previous input values and current temperature to randomize the weights, it depends on the type of problem we are going to solve with trained neural network. In this paper, we have used TANH as an activation function so we have to normalize the input, output and their corresponding weights in between (-1,1) and a weight matrix is created which acts as a linear array of double data type. The randomization of weights is not a complex task, in this paper we have generated a random number and multiply it with current temperature.

Q = currentTemp * Random (N)

Then this number is multiplied by each value in the weight matrix Wij*Q and updates all the values. This task is performed in each 'n' cycles and the updated matrix is compared with the previous one, if it is better than previous then the weight matrix gets updated.

V. RESULTS

In this section, we present the result which we get by using this algorithm on real world dataset. The dataset is taken from UCI repository consist of 1000 instance. This dataset contains useful information about transaction. This dataset contains 20 attributes and values corresponding to each rows is converted into symbolic or numerical form because of some privacy agreement. But the types of attributes is mentioned, shown in the table I.

TABLE I: ATTRIBUTES USED FOR TRAINING AND EVALUATING THE NEURAL NETWORK

S. No	Attribute Name
1	Status of existing checking Account
2	Duration in month
3	History of credit taken
4	Purpose of transaction
5	Credit amount
6	Saving account/bonds
7	Present employment since
8	Instalment rate in %
9	Personal status & sex
10	Guarantors
11	Present address since
12	Property
13	Age
14	Other instalment plans
15	Housing
16	Existing credits at this bank
17	Job
18	Number of people liable to provide maintenance for
19	Telephone
20	Foreign worker(Yes or No)

In this paper, we have divided the dataset into two parts: 75% data is taken from dataset for the training purpose and 25% data is taken for the evaluation of trained neural network. So, the evaluation dataset is new for the trained neural network, by performing the evaluation task we can classify the unseen data as a fraud or non-fraud. In this paper we present the result by showing the percentage of correct & incorrect classified data by comparing the predicted label with the existing label.
The configuration of Neural Network and the list of parameter used in Simulated Annealing to train neural network and their respective values is shown in table 2 & 3 respectively.

TABLE II. PARAMETERS OF AN ARTIFICIAL NEURAL NETWORK

Input layer Neurons	Hidden layer Neurons	Output layer neurons	Activation Function
20	50	2	Hyperbolic Tangent Activation Function

TABLE III. PARAMETERS OF SIMULATED ANNEALING ALGORITHM TO TRAIN NEURAL NETWORK

Start Temperature	Stop temperature	Number of Cycles/Iteration
100	3	100

After running this program it takes almost two days and stop training after reaching 1% error. After the process of training is, it goes for the evaluation process. This project is based on real-time i.e. at the time when performing any transaction at any online portal, so the time taken to classify any unseen data should be

less. We have implemented this algorithm in java and perform an evaluation on the 25% of the dataset. It classifies all the data i.e. 250 instances, within 5-10 seconds, which is good in comparison with different configuration of neural network.

TABLE IV. RESULT OF TRAINED NEURAL NETWORK WHEN EVALUATION DATASET IS APPLIED

Observed	Total	Correct	Percentage Correct
Total Case	250	224	89.6%
Fraud Case	173	159	92%
Non-Fraud Case	77	65	85%

Table IV display the result of the trained neural network using simulated annealing. As we can see, in the first row the total case is only 250 instances from which 224 data is correctly classified i.e. based on the pre-defined label, the label predicted by the trained network is correct.

VI. CONCLUSION

In this paper we showed that better result is achieved with ANN when trained with simulated annealing algorithm. As the result shows that the training time is high but the fraud detection in real time is considerably low and the probability of predicting the fraud case correctly in online transaction is high, which is a main measure to evaluate any ANN. In the table 3 we can see that 65% of total fraud case is correctly classified which is a very high percentage in comparison with genetic, resilient backpropogation and any other training algorithm.

The main problem in credit card fraud detection is the availability of real world data for the experiment. This approach can also be used in other applications which require classification task [20] e.g. software failure prediction, etc.

A. Future Work in this project

There will be a lot of work to be done for fraud detection because the activity of user is different in each transaction which causes the training of any ANN to be difficult. In this project the main task is to find the best configuration for neural network, we can use Genetic Algorithm for this task, it would find a better configuration by applying different combinations. So if we combine Simulated Annealing and Genetic Algorithm to create a best model, it will gives better result than any other.

REFERENCES

[1] Linda Delamaire (UK), Hussein Abdou (UK), John Pointon (UK),"Credit card fraud and detection techniques: a review", Banks and Bank Systems, Volume 4, Issue 2, 2009.

[2] Nadeem Akhtar, Farid ul Haq, " Real Time Online Banking Fraud Detection Using Loaction Information",International Conference on Computational Intelligence and Information Technology – CIIT 2011, Pune, India.

[3] K. Cios, W. Pedrycs, and R. Swiniarski, Data Mining Methods for Knowledge Discovery. Boston: Kluwer Academic Publishers, 1998.

[4] Y. Sahin and E. Duman,"Detecting Credit Card Fraud by Decision Trees and Support Vector Machines", International conference of Engineers & computer Scientists 2011 Vol I, March 16 2011, Hong Kong.

[5] Bolton, R. J. and Hand, D. J.,"Statistical fraud Detection: A review". Statistical Science 28(3):235-255, 2002.

[6] Karl BlomStorm," Benchmarking an artificial neural network tuned by a genetic algorithm", VT 2012.

[7] UCI Machine Learning Repository,"http://archive.ics.uci.edu/ml/datasets.html", last accessed at 22/11/2013.

[8] Wai-cgiu Wong, Ada Wai-chee Fu,"Incremental Document Clustering for Web Page Classification", Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Springer Japan 2002.

[9] W. Wong and A. Fu, "Incremental Document Clustering for Web Page Classification," Proc. 2000 Int'l Conf. Information Soc. in the 21st Century: Emerging Technologies and New Challenges (IS2000), 2000.

[10] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", Psychological review, 65(6):386, 1958.

[11] Han, Jun; Morag, Claudio," The influence of the sigmoid function parameters on the speed of Backpropagation learning", In Mira, José, Sandoval, Francisco, From Natural to Artificial Neural Computation. pp. 195–201, 1995.

[12] Metropolis, Nicholas, Rosenbluth, Arianna W., Rosenbluth, Marshall N., Teller, Augusta H., Teller, Edward, "Equation of State Calculations by Fast Computing Machines", The Journal of Chemical Physics 21 (6): 1087, 1953.
[13] Kirkpatrick, S., Gelatt Jr, C. D., Vecchi, M. P., "Optimization by Simulated Annealing". Science 220 (4598): 671–680, 1983.
[14] Cerny, V., "Thermo dynamical approach to the traveling salesman problem: An efficient simulation algorithm", Journal of Optimization Theory and Applications 45: 41–51, 1985.
[15] P J van Laarhoven and E H Aarts,"Simulated Annealing: Theory and Applications", Kluwer Academic Publishers, 1987.
[16] R H Otten and L P Ginneken,"The Annealing Algoritm", Kluwer Academic Publishers, 1989.
[17] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu, "Learning Approaches for Detecting and Tracking News Events," IEEE Intelligent Systems, vol. 14, no. 4, pp. 32-43, 1999.
[18] Jonathan Engel," Teaching Feed-Forward Neural Networks by Simulated Annealing", Norman Bridge Laboratory of Pllysics 161-33, California Institute of Technology, Pasadena, CA 91125, USA Complex, Systems 2, 1988.
[19] Mohamed Benaddy and Mohamed Wakrim,"Simulated Annealing Neural Network for Software Failure Prediction", International Journal of Software Engineering and Its Applications Vol.6, No. 4, October, 2012.
[20] Li, Y.H., Jain, A.K.: Classification of Text Documents. The Computer Journal. vol. 41, pp. 537--546 (1998).

Egyptian Computer Science Journal (ISSN: 1110 – 2586)
Volume 40 – Issue 03, September 2016

Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques

Mohamed Hegazy[1], Ahmed Madian[2, 3], Mohamed Ragaie[1]
[1]Information System department the Arab Academy for Science Technology and Maritime Transport
[2] Nano-Electronics Integrated Systems Center, Nile University, Cairo, Egypt
[3] National Center for Radiation Research, Egyptian Atomic Energy Authority, Egypt
hegazy_prog@yahoo.com, ah_madian@hotmail.com, ragaie2@mcit.gov.eg

Abstract

This paper aimed to build unified pattern per customer not only represent normal behavior but also Fraud pattern that's represented previously and confirmed as fraud transactions that's facilitate studding fraudsters behavior. An enhancement for the proposed algorithm of Fraud Miner has been proposed. This enhancement involves introducing LINGO clustering Data mining algorithm by replacing Apriori algorithm used in Fraud Miner for Frequently Pattern creation and facilitate summarize customer previous behavior either within his Legal or Fraud transactions. Using this algorithm provide more chance for easily fraud detection as the fraudsters always behaving same as customer behaviors instead of study fraudster behavior the customer frequent behavior will be identified from his legal or previously confirmed transactions being fraud. A performance comparison with other algorithms has been carried out.

Keywords*: Apiriori, Clustering data mining, Fraud detection, Lingo, PCI-DSS.*

1. Introduction

The innovation of new technologies, communication techniques and from the fact of Fraud is ubiquitous; it does not discriminate in its occurrence. Anti-fraud controls can effectively reduce the likelihood and potential impact of fraud, actually no entity is immune to this threat. Unfortunately, however, many organizations still suffer from an "it can't happen here" mindset [1].

Valuable knowledge and interesting patterns are hidden in this data. There are huge potential for banks to apply data mining in their decision making processes in areas like marketing, credit risk management, and detection of money laundering, liquidity management, investment banking and detection of fraud transactions in time Failures in these areas can lead to unpleasant outcomes for the bank such as losing customers to competition, financial loss, reputational loss and hefty fines from the regulators. According to Review provided in [2] about data mining we have two common used approaches for fraud detection in banking industries as shown in Figure 1.

Users go through reports generated by banking information system and use it in their decision making process. Manual analysis has limitations because volumes of data that can

be manually analyzed are limited and hence the decisions may not be as accurate as intended.

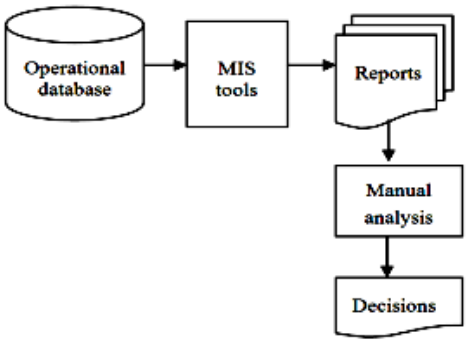


Figure 1. Conventional decision making process

It is assumed that valuable information are hidden in this volume of operational and historic data that can be used for critical decision making process if they are discovered and put to use by capable tools [3]. For example, a decision support system based on data mining techniques can be employed to improve the quality of lending process in a bank [4]. The second recommended approach that showed in Figure 2 how data mining can improve decision making process.

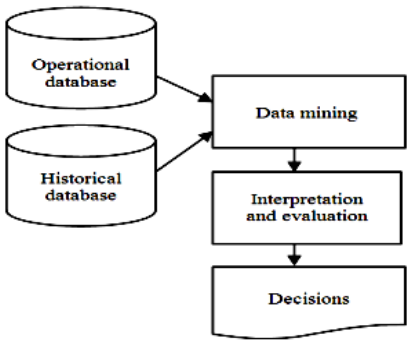


Figure 2. Decision making with data mining

According to ACFE (Association of Certified Fraud Examiners) report of 2014 that contains analysis of 1483 cases that caused in excess of $ 3 billion in losses due to fraud as the median loss caused by a single case of occupational fraud is $ 145,000 that record that financial statement fraud schema type represent high median losses that mainly appeared on Banking and financial services industry that we have the main focused here specially Credit card fraud is practiced most frequently amongst the varied financial frauds due to its acceptance and widespread usage as it offers more convenience to its users.

Due to the nature of huge amount of transactions that need to be manually analyzed which are limited that negatively impacts the decisions accuracy , Data Mining techniques as One domain data mining can excel at, suspicious transaction monitoring, has emerged for the first time as the most effective fraud detection method in 2011 according to Survey [5]. Out of the available data mining techniques, clustering has proven itself a constant applied solution for detecting fraud. Anomaly detection in the past couple of years achieved good results in knowing customers' pattern that’s detects any Out-Of-Norm activities.

This paper proposes Credit card fraud detection model that's handle imbalanced dataset and facilitate knowing of customers' patterns by splitting data into legal (confirmed True transactions) and fraud (Confirmed Fraudster behaviours) patterns to eliminate the problem of imbalanced dataset.

Paper organized as follows Section 2 provides the innovation of enhanced fraud Miner and introduces the usage of LINGO clustering Data mining algorithm.

Section 3 explains the methodology and techniques approached for implementation. Section 4 involves testing of proposed model and results discussion specially, when compared with the original algorithm of fraud Miner. Section 5 provides conclusion and Future work.

2. Background and Related Work

Many security controls implemented by VISA and MasterCard on transactions level that reflected either from PCI-DSS (Payment Card Industry Data Security Standards) set of rules and policies or from the force of using the new chip cards that uses EMV (Euro pay MasterCard VISA) technology [6] that solves many security vulnerabilities appeared on old magnetic strip cards some of these vulnerabilities related to card skimming that consider type of card fraud which is involves the theft of credit card information used in an otherwise legitimate transaction as EMV achieved great efforts concerning contactless operations [7] whereas most of Europe's cards are Contact-based as EMV facilitate offline authorization With ISO 14443 becoming a payment standard and merchants are able to accept contactless payments from different card companies, including Visa and MasterCard that reflect on many convenience to merchants.

Comparative study for different biometric authenticators technologies that might be used in online banking [8],study proved that fingerprint, iris and face are the most appropriate for inclusion in biometric authentication systems of online banking specially when two-factor authentication are required.

Credit card transactions trained using Baum-Welch algorithm in [9] by modeling sequence of operation using Hidden Markov Model (HMM) and dividing transactions into three groups high, medium and low according to transaction amount so that spending profile of cardholder created more easier.

Hybrid algorithm proposed in [10] for credit card fraud detection based on combination of Naïve Bayes algorithm with Hidden Markov model and offering OTP (One Time Password) for newly transactions for more security about newly behaviours.

Principle component analysis proposed in [11] aimed to represent each sample of transaction with few number of values so that attributes could be reduced by determine attributes contains major information and facilitate faster fraud detection for credit card transactions.

Novel web clustering Data Mining invented in [12] which considered as strong emphasis is placed on the high quality of group descriptions named by LINGO algorithm based on Latent Semantic Indexing and Singular Value Decomposition that's firstly identify good cluster labels with meaningful meaning then assigning the contents to each label. LINGO produces reasonably described and meaningful clusters when implemented into the Carrot2 framework that significantly influence the quality of clustering.

LINGO algorithm implemented in [13] and proposed this algorithm as an approach for clustering could be used for outlier detection as the proposed algorithm idea is to first find meaningful descriptions of clusters (Description come First "DCF") then assign documents to the produced labels using (LSI) Latent Semantic Indexing and regarding documents assigning Vector Space Model used to determine cluster content.

Many studies and efforts spend in credit card fraud detection, K-Mean proposed in [14] as clustering data mining algorithm to indicate legal/ fraud transactions however real data seems not available but proposed algorithm showed significant results in fraud detection.

This work aimed to enhance current Fraud Miner algorithm provided by [15] that proposed new fraud detection technique to handle imbalance class by identifying fraud patterns for each customer instead of finding a common pattern for fraudulent behaviour.

Using data Pre-classification that determine legal and fraud transactions per customer to facilitate speed up the process of fraud detection as both legal and fraud behaviour doesn't have frequently changes, it's changes over longer period of time.

Applying Apriori algorithm to generate legal and fraud patterns then using customized matching algorithm transaction fraud detection process become more easier and enables real time detection ,it's recorded highest fraud detection rate and showed good performance in handling class imbalance when compared to NB(Naive Bayes) , SVM (Support Vector Machine) , RF(Random Forest), KNN (K-Nearest Neighbour) classifiers.

It's noticed that the majority of clustering mining algorithms made the discovery process first then based on contents labels inducted that some time result in some groups' description meaningless as this problem solved in [12] by introducing Lingo as radically different approach to finding and describing groups that aimed to firstly find meaningful cluster description then assigns snippers to them and introduce DCF(Description Comes First) method using Singular Value decomposition(SVD) as reduction technique [13] as indicated below summarized steps:

(a) Data Pre-processing by apply stemming, text filtering and remove stop words.
(b) Feature Extraction that aimed to discover frequent items and phrases.
(c) Cluster Label induction that's find best matching phrase.
(d) Cluster Content Discovery that's assign contents to the resulted clusters.
(e) Final Cluster formation that's Calculate cluster scores and apply cluster merging.

Lingo algorithm implemented as a component of Carrot2 framework that achieved good results, Apriori algorithm have the lowest memory usage when compared with different algorithms of association rules [16], Apriori requires many database scan till having a refined pattern that's negatively affect the performance due to consuming time [17] which reduce the performance of fraud, This paper implements Lingo algorithm for valid/fraud pattern creation instead of Apriori algorithm that proposed in fraud Miner[15].

3. Methodology

The proposed enhancement for Fraud Miner will involve Phases as follows:

3.1 Data Preparation

Due to sensitivity and confidentiality of needed card holder data required for test and Banks limitation to provide this data for test so before starting Data preparation phase it's required to have transactions simulator that responsible for simulate transactions and prepare

appropriate imbalanced dataset.

Data Pre-processing would be required after dataset formulation as follows:

(a) Refine data by Remove the transactions corresponding to those customers who have only one transaction in dataset.

(b) Segregate transactions into legal and fraud transactions.

The refined imbalanced data represented in Table 1.

Table 1. Imbalanced data number of transactions in training set

Number of customers	Legal	Fraud	Total
200	*40225*	*12236*	*52461*
400	*60957*	*14075*	*75032*
600	*101600*	*16050*	*117650*
801	*133526*	*20522*	*154048*
1000	*165841*	*24372*	*190213*
1200	*209097*	*26817*	*235914*
1400	*241657*	*31126*	*272783*

3.2 Algorithms Implementation and Patterns Creation

Prepare an Implementation for Apiriori and Lingo algorithm according to the nature of simulated test data we have below lingo attributes should be set within Lingo algorithm:

(a) Set desired Cluster Count Base to 25 as this attribute refer to desired cluster count base as a Base factor used to calculate the number of clusters based on the number of documents on input. The larger the value, the more clusters will be created. The number of clusters created by the algorithm will be proportional to the cluster count base, but not in a linear way.

(b) Set cluster Merging Threshold value to be 0.9 as this attribute refer to Cluster merging threshold. The percentage overlap between two cluster's documents required for the clusters to be merged into one clusters. Low values will result in more aggressive merging, which may lead to irrelevant documents in clusters. High values will result in fewer clusters being merged, which may lead to very similar or duplicated clusters.

(c) Set Stop Word Label Filter. enabled false instead of disabled as this attribute intend to Remove stop labels. Removes labels that are declared as stop labels in the stop labels. <lang> files. Please note that adding a long list of regular expressions to the stop labels file may result in a noticeable performance penalty.

(d) Set Document Assigner.min Cluster Size 1 instead of default 2 that’s determines the minimum number of documents in each cluster.

Run Apiriori Pattern Generation application to generate customers' related Fraud and Legal patterns using Apiriori algorithm then run LINGO pattern Generation application to generate customers' related Fraud and Legal patterns using LINGO algorithm.

4. Proposed Algorithm Efficiency

This section aimed to test proposed algorithm efficiency when compared to old fraud miner and discuss output results.

4.1 Results

Due to the imbalanced nature of data we have 4 classification metrics relevant to credit card fraud detection measures [18] fraud detection rate, false alarm rate, balanced classification rate, and Matthews's correlation coefficient.

Here, fraud is considered as positive class and legal as negative class and hence the meaning of the terms P, N, TP, TN, FP, and FN are defined as follows:

Positives (P): number of fraud transactions;
Negatives (N): number of legal transactions;
True positives (TP): number of fraud transactions predicted as fraud;
False negatives (FN): number of fraud transactions predicted as legal.
True negatives (TN): number of legal transactions predicted as legal;
False positives (FP): number of legal transactions predicted as fraud;

Before starting the comparison we need to prepare data by develop program that's get fraud and legal transactions per table count and pass every single transaction as incoming transaction to verify and by implementing matching algorithm provided in [15] to detect whether the incoming transaction fraud or legal and put results in summary as represented in Table 2 sample of legal transactions test.

Table 2. Sample of Legal Transactions

Working algorithm	Customers count	Fraud count	Legal count
Apiriori	200	51	149
Apiriori	400	79	321
Apiriori	600	118	482
Apiriori	801	123	678
Apiriori	1000	123	877
Apiriori	1200	123	1077
Apiriori	1400	123	1277
Lingo	200	44	156
Lingo	400	67	333
Lingo	600	100	500
Lingo	801	102	699
Lingo	1000	102	898
Lingo	1200	112	1088
Lingo	1400	106	1294

The performance of the enhanced algorithm evaluated with the original fraud miner proposed in [15] as from the Fraud miner performance evaluation previously held with other four credit card fraud detection algorithms support vector machine (SVM), K-Nearest

neighbour classifier, naıve Bayes classifier, and random forest, Fraud Miner were having highest fraud detection rate than other classifiers with very less false alarm rate.

Fraud Detection Rate is the percentage of correct positive fraud transactions from actual Fraud transactions.

Figure 3 shows the performance of Enhanced fraud Miner that's represented the same performance of the original Fraud Miner.

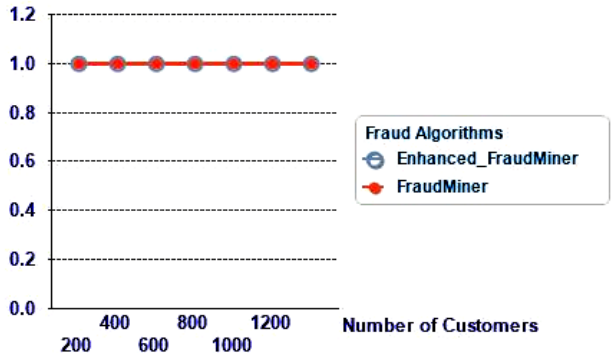


Figure 3. Sensitivity Rate

False Alarm Rate. Which represent number of actual negatives transactions predicted as positives.

Figure 4 shows the performance of enhanced Fraud Miner on False alarm rate that's represented more reducing in false alarm rate when compared with the original Fraud Miner.

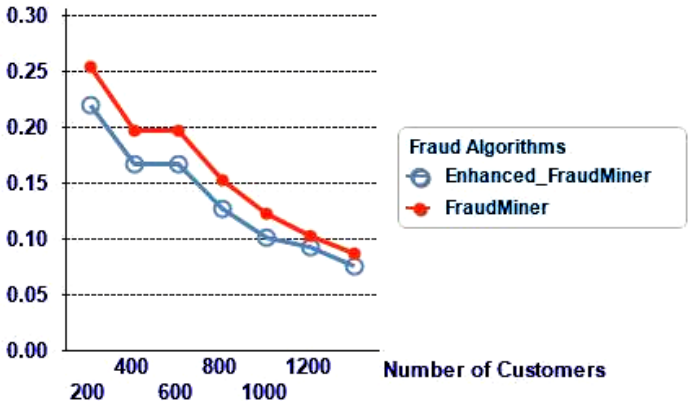


Figure 4. False Alarm Rate

Balanced Classification Rate (BCR). Which represent the average of sensitivity and specificity as Figure 5 represented small enhancement in this rate for the enhanced fraud Miner.

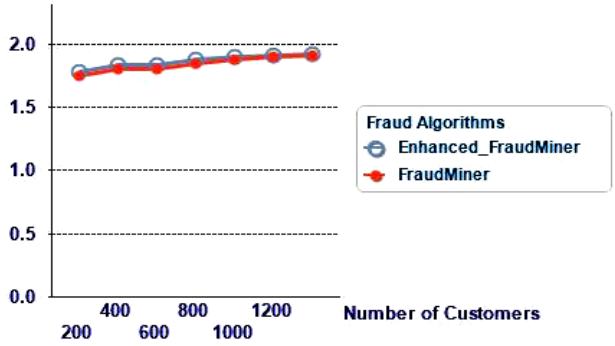


Figure 5. Balanced Classifier Rate

Matthews correlation coefficient (MCC). Which is used as a measure of the quality of binary classifications as according to nature of simulated test data and from Figure 6 it's found that enhanced Fraud Miner have some enhancement in quality of binary classifier.

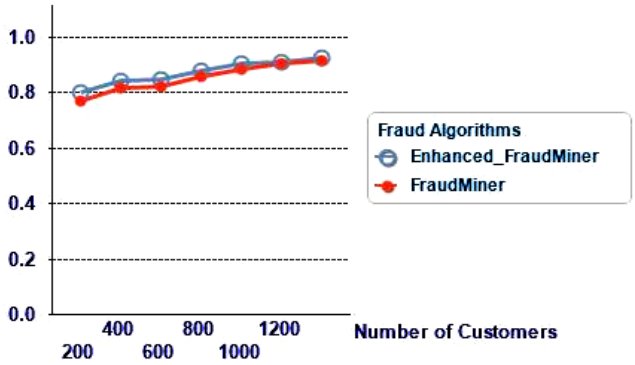


Figure 6. Mathews Correlation Coefficient

4.2 Discussion

Measuring Fraud detection algorithm require to pay more attention for sensitivity and False alarm rate, Fraud Miner recorded highest fraud detection and lowest false alarm rates when compared to other classifier. In proposed enhanced algorithm (Figures 3 and 4) we have not only output the same sensitivity but also decreasing false alarm rate and improving customer satisfaction.

Quality of algorithms that handled imbalanced data measures using balanced metrics of BCR and MCC, A coefficient of +1 means highest quality, 0 means algorithm act like random prediction system, and -1 means very low quality and algorithm failed in prediction and observation. Proposed enhanced algorithm (Figures 5 and 6) shows the same quality achieved in fraud Miner with some coefficient enhancements that's increases prediction.

Using the same matching algorithm in the proposed enhanced fraud miner resulted in keeping limitation within fraud detection, especially in case of identical transactions exist both in Legal and Fraud patterns (overlapping) that's leads to unable to recognize fraud transactions.

5. Conclusion and Future Work

In this paper, a Survey on different Data Mining techniques used for Credit card fraud detection has been introduced. Fraud/Legal Pattern creation for each customer facilitate customer profile detection not only normal behaviour but also fraudster behaviors on his account and this made fraud detection easier. Also, the LINGO algorithm proved to be used to create Legal/Fraud patterns per each customer instead of Apiriori Algorithm and work almost in the similar efficient. By using simulated test transactions it's found that LINGO generate meaningful summarized patterns more than output patterns from Apriori Algorithm. Fraud/Legal Pattern creation facilitate fast of fraud detection process and could be used to verify transaction near real time transactions.

According to comparison results of proposed model for enhancing Fraud Miner algorithm it's achieved good enhancements especially with the high important measure of False alarm rate that decreased more in the proposed model also it's noticed an improvements of algorithm quality that represented from Matthews correlation coefficient.

The future work suggested to apply LINGO3G as enhanced LINGO algorithm that have many improvements like achieving very fast clustering over huge snippets records, improving cluster label quality , Hierarchical Clustering , promoting specific words or phrases in the output cluster labels and defining groups of words or phrases to be treated as synonymous with advanced Results tuning.

Credit card transactions could be segregated according to transaction source so that group ATM, POS, Merchant Draft Entry (MDE) and other types of transactions to facilitate more speed in fraud detection process.

References

[1]. Singleton T. W.: Fraud Auditing and Forensic Accounting 4th edition, Ed. John Wiley and Sons, (2010)

[2]. Pulakkazhy, S., &Balan, R. V. S.: Data Mining in Banking and Its Applications-a Review. Journal of Computer Science, 9(10), pp. 1252–1259. doi:10.3844/jcssp.2013.1252.1259, (2013)

[3]. Kazi, I.M. and. Q.B. Ahmed: Use of data mining in banking. Int. J. Eng. Res. Appli., pp. 738-742 (2012)

[4]. Ionita, I. and L. Ionita: A decision support based on data mining in e-banking. IEEE Preccedings of the 10th Reodunet International Conference (RoEduNet), Jun. 23-25, IEEE Xplore Press, Iasi, pp. 1-5. DOI: 10.1109/RoEduNet.2011.5993710 (2011)

[5]. Sorin, A.: Survey of Clustering based Financial Fraud Detection Research. InformaticaEconomica, 16(1), pp. 110–123, (2012)

[6]. EMV-Book3 Card, I. C. vol. 3, (June 2008)

[7]. Greenemeier, L., : Visa expands contactless card efforts, InformationWeek, March 27, http://tinyurl.com/ykzo4t (2006)

[8]. Parusheva, S. "A comparative study on the application of biometric technologies for authentication in online banking", Egyptian Computer Science Journal (ECS), Vol.39, No.4, ISSN-1110-2586, September 2015, pp.115–126.

[9]. Matheswaran, P., Me, E. S. S., & Rajesh, R. (2015). Fraud Detection in Credit Card Using DataMining Techniques, II(I), 11–18

[10]. Gupta, A., &Raikwal, J.: Fraud Detection in credit Card Transaction Using Hybrid Model, 3(1), pp. 3730–3735, (2014)

[11]. D. Pawar, A., N. Kalavadekar, P., & N. Tambe, S.: A Survey on Outlier Detection Techniques for Credit Card Fraud Detection. IOSR Journal of Computer Engineering, 16(2), pp. 44–48. doi:10.9790/0661-16264448 , (2014)

[12]. Osi'nski, S.: An algorithm for clustering WEB SEARCH RESULTs. Journal of Mathematical Psychology, 12(3), pp. 328–383 ,(2003)

[13]. Fafat, P. C., &Sikchi, P. S. S. :Lingo an approach for Clustering, 1(3), pp. 1–3 , (2012)

[14]. Journal, I., Applications, C., & Design, V. :Fraud Detection in Credit Card by Clustering Approach, 98(3), pp. 29–32 (2014)

[15]. Seeja, K. R., &Zareapoor, M. FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining. TheScientificWorldJournal 2014,252797. doi:10.1155/2014/252797, (2014)

[16]. Alhamzi, A., Nasr, M., &Salama, S. "A Comparative Study of Association Rules Algorithms on Large Databases", Egyptian Computer Science Journal (ECS), Vol.38, No.3, ISSN-1110-2586, September 2014, pp.51–62.

[17]. Mohammed Al-Maolegi, B. A.: An improved apriori algorithm for association rules of mining. International Journal on Natural Language Computing, 3(1), pp. 942–946. doi:10.1109/ITIME.2009.5236211, (2014)

[18]. François, D.: Binary classification performances measure cheat sheet. Journal of Machine Learning Research, 7, pp. 1–30, (2006).

mathematics

MDPI

Article

Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest

Tzu-Hsuan Lin [1] and Jehn-Ruey Jiang [2,*]

[1] Department of Computer Science, University of Southern California, Los Angeles, CA 90007, USA; lintzuhs@usc.edu
[2] Department of Computer Science and Information Engineering, National Central University, Taoyuan City 320317, Taiwan
* Correspondence: jrjiang@csie.ncu.edu.tw

Abstract: This paper proposes a method, called autoencoder with probabilistic random forest (AE-PRF), for detecting credit card frauds. The proposed AE-PRF method first utilizes the autoencoder to extract features of low-dimensionality from credit card transaction data features of high-dimensionality. It then relies on the random forest, an ensemble learning mechanism using the bootstrap aggregating (bagging) concept, with probabilistic classification to classify data as fraudulent or normal. The credit card fraud detection (CCFD) dataset is applied to AE-PRF for performance evaluation and comparison. The CCFD dataset contains large numbers of credit card transactions of European cardholders; it is highly imbalanced since its normal transactions far outnumber fraudulent transactions. Data resampling schemes like the synthetic minority oversampling technique (SMOTE), adaptive synthetic (ADASYN), and Tomek link (T-Link) are applied to the CCFD dataset to balance the numbers of normal and fraudulent transactions for improving AE-PRF performance. Experimental results show that the performance of AE-PRF does not vary much whether resampling schemes are applied to the dataset or not. This indicates that AE-PRF is naturally suitable for dealing with imbalanced datasets. When compared with related methods, AE-PRF has relatively excellent performance in terms of accuracy, the true positive rate, the true negative rate, the Matthews correlation coefficient, and the area under the receiver operating characteristic curve.

Keywords: autoencoder; credit card; deep learning; fraud detection; data imbalance; random forest

Citation: Lin, T.-H.; Jiang, J.-R. Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest. *Mathematics* **2021**, *9*, 2683. https://doi.org/10.3390/math9212683

Academic Editors: Radu Tudor Ionescu and Guansong Pang

Received: 12 September 2021
Accepted: 21 October 2021
Published: 22 October 2021

1. Introduction

Credit card fraud is the unauthorized use of credit cards to obtain money, goods, or service by fraud. With the rise of e-commerce and contactless payment, credit cards are now widely used anywhere and anytime. For example, there are an estimated 1.1 trillion credit cards in the United States alone [1]. Thus, it is not surprising that millions of people fall victim to credit card fraud every year. Due to the prevalence of credit card transactions, cases of credit card fraud are also rampant globally. Americans reported 271,823 credit card fraud cases in 2019, an increase of 72.4% from 2018 [2]. Monitoring credit card transactions is not easy due to the large volume of data. Therefore, credit card fraud transactions are easily ignored, leading to huge losses for both cardholders and issuers. According to Nilson Report [3], credit card fraud caused losses of USD 28.65 billion in 2019, increasing by 2.9% from USD 27.85 billion in 2018. By 2020, global financial losses caused by credit card fraud amounted to USD 31 billion [4]. In addition to cardholders' vigilance and issuers' supervision, effective fraud detection methods must be adopted to detect credit card fraud automatically. This motivates the authors to develop credit card fraud detection methods based on advanced technologies.

Many credit card fraud detection methods [2,4–15] have been proposed in the literature. The readers are referred to two survey papers [16,17] for detailed descriptions of the methods. The survey paper [16] raised three challenging problems in credit card fraud

detection. The first is the data imbalance problem caused by the huge difference between the numbers of the positive and the negative classes. Specifically, as normal transactions far outnumber fraudulent transactions, credit card fraud detection methods are likely to overfit normal transactions. The second problem is the dataset shift, which means that fraud behaviors may evolve. New customer behaviors and new attacks on credit card transactions will deter fraud detection methods from maintaining good performance. The last problem is the oversight of sequential information among adjacent transactions. This is because investigators usually focus on a separate transaction but features of a separate transaction cannot reveal relations hidden among adjacent transactions. Different approaches are proposed to address different problems mentioned above. For instance, the papers [5,6] detailed the data shift problems and gave corresponding solutions to this problem. As for the ignorance of sequential information, it was tackled in the papers [7,8], which proposed methods having been tested to be effective. Some papers [9–12] utilized data resampling mechanisms to solve the data imbalance problem to have good fraud detection performance. However, some other papers [13–15] proposed methods that are naturally suitable for dealing with the data imbalance problem.

This research proposes a method, autoencoder with probabilistic random forest (AE-PRF), for credit card fraud detection. The proposed AE-PRF method first uses the autoencoder (AE) [18] to extract transaction data features. It then employs the random forest (RF) [19] with probabilistic classification to classify credit card transactions as normal or fraudulent. As just mentioned, the AE and the RF models can efficiently handle imbalanced data [20,21]. AE-PRF adopts the AE and the RF models since credit card transactions are typical imbalanced data. Moreover, unlike other methods adopting the RF with 0/1 classification, AE-PRF adopts the RF with probabilistic classification so that the performance of AE-PRF can be further improved, as will be shown later.

The credit card fraud detection (CCFD) dataset [22] released on the Kaggle platform was applied to AE-PRF for performance evaluation. The CCFD dataset contains credit card transactions of European cardholders within two days, including the normal transactions and the fraudulent transactions. It is extremely imbalanced, as the fraudulent data account for only 0.172% of total data. To make the CCFD dataset more balanced, data resampling schemes such as the synthetic minority oversampling technique (SMOTE) [23], adaptive synthetic (ADASYN) [24], and Tomek link (T-Link) [25] were applied to CCFD before data were fed into AE-PRF. As will be shown later, the performance of AE-PRF did not vary much whether resampling schemes were applied to the dataset or not. This indicates that AE-PRF is naturally suitable for dealing with imbalanced data. The performance of AE-PRF was compared with those of most related methods [12–15] that rely on CCFD for performance evaluation. Note that the methods proposed in [12] take or do not take data resampling, whereas the other methods proposed in [13–15] do not take data resampling. The performance comparisons are shown in terms of the accuracy, true positive rate, true negative rate, Matthews correlation coefficient, and area under the receiver operating curve to show the superiority of AE-PRF.

The contribution of the paper is threefold. First, it proposes AE-PRF that first uses AE to extract features of low-dimensionality from credit card transaction data features of high-dimensionality, and then relies on the RF with probabilistic classification to classify data as fraudulent or normal. By adopting the RF with probabilistic classification, the performance of AE-PRF can be improved. Second, experiments were conducted to apply data resampling schemes to imbalanced data before they were fed into AE-PRF. The experimental results show that AE-PRF is naturally suitable for dealing with imbalanced data, as the performance of AE-PRF does not vary much whether resampling schemes are applied to the data or not. Third, extensive experiments were conducted to evaluate the performance of AE-PRF and the performance evaluation results were compared with those of existing methods proposed in the literature [12–15]. The comparison results show that AE-PRF has relatively excellent performance in terms of accuracy, the true positive rate,

the true negative rate, the Matthews correlation coefficient, and the area under the receiver operating characteristic curve.

The rest of this paper is organized as follows. Section 2 describes the proposed AE-PRF method and some preliminaries. Section 3 then details related work. The performance evaluation of AE-PRF and its comparisons with related methods are shown in Section 4. Finally, Section 5 concludes this paper.

2. The Proposed Method

As just mentioned, the proposed AE-PRF method uses the AE to extract data features and employs the RF with probabilistic classification to classify credit card transactions as normal or fraudulent. To describe AE-PRF clearly, the concepts of the AE and the RF are first elaborated below.

2.1. Autoencoder

An AE [18] is a special type of artificial neural network that comprises connected neurons. Each neuron takes input vector x and generates the output y according to the following Equation (1):

$$y = \sigma\left(wx^{\mathrm{T}} + b\right), \tag{1}$$

where $\sigma(\cdot)$ is a nonlinear activation function (e.g., a sigmoid function), w is a weight vector, x^{T} is the transposition of x, and b is a bias vector.

The neural network structure of an AE is symmetric, as shown in Figure 1. An AE has one input layer, one or more hidden layers, and one output layer. Especially, the output layer of an AE has the same number of neurons as the input layer. Furthermore, the kth hidden layer and the $(n-k+1)$th hidden layer (or the kth hidden layer from the bottom) have the same number of neurons, where $k = 1, \ldots, \lfloor n/2 \rfloor$, and n is the number of hidden layers. The middle hidden layer is called the bottleneck, and the states (values) of neurons in the bottleneck layer constitute the code or the latent representation of the input. The code can be regarded as the extracted feature or the dimensionality reduction result of the original input.

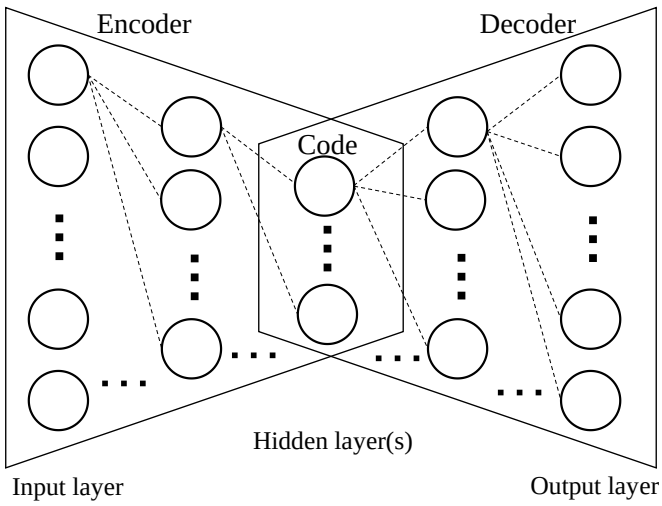


Figure 1. The neural network structure of an autoencoder (AE) model.

The first half part of an AE is called the encoder, whereas the second half part is called the decoder, as shown in Figure 1. The encoder encodes the input into the code, and the decoder decodes the code into the output. The output is intended to be as close to the input as possible; it is called the reconstructed input. The difference between the input and the output is called the reconstruction error. The AE is trained with the goal of minimizing the reconstruction errors by using the error backpropagation, gradient descent, and various optimizers like adaptive moment (Adam) optimizer.

2.2. Random Forest

The RF is an ensemble learning model for classification, regression, and other tasks [19]. Since the proposed AE-PRF method is for the task of classification, only the classification task is discussed in the following context. Specifically, the RF model utilizes decision trees to classify data and employs the bagging (i.e., bootstrap aggregating) approach to avoid the overfitting problem caused by complex decision trees. Below, the concept of using decision trees to classify data is first described.

A decision tree is a tree-like structure in which each internal node has a "split" based on an attribute, and each leaf node represents a prediction (or classification) result. Some metrics, such as the Gini impurity, entropy, and standard deviation, can be used for selecting the best splitting with the largest information gain. Below, the Gini impurity is taken as an example to show how the information gain is measured in decision trees. The information gain $IG(N_p, a)$ at node N_p split into c child nodes $N_1, \ldots, N_c$ based on the attribute a is defined in the following Equations (2) and (3):

$$IG(N_p, a) = Gini(N_p) - \sum_{i=1}^{c} \frac{|N_i|}{|N_p|} Gini(N_i) \tag{2}$$

$$Gini(N_p) = 1 - \sum_{j=1}^{m} p_j^2 \tag{3}$$

In Equation (2), $|N_p|$ stands for the number of data at node N_p, and $|N_i|$ stands for the number of data at node N_i, $0 \leq I \leq c$. In Equation (3), m is the number of different labels of data at node N_p, and p_j is the ratio of the number of data with the jth label over the total number of data at node N_p.

The best splitting with the largest information gain is performed for every possible attribute and every possible attribute value of dividing. The splitting continues until one of the following three stop conditions occurs. The three stop conditions are (i) all data at a node have the same label, (ii) the number of data at a node reaches a pre-specified minimum limitation, and (iii) the depth of a node reaches a pre-specified maximum limitation. After the splitting stops, the decision trees can be used to classify an input sample. The input sample goes through the tree from the root node to a leaf node, and it is classified as the label that dominates others at the leaf node.

Below we describe the bagging approach that randomly selects partial data and partial attributes to construct a variety of decision trees to be combined for data classification. This can avoid the overfitting problem that is intrinsic in decision trees. Given a dataset of d data or observations, the bagging approach produces n sub-datasets by drawing d' out of d observations with replacement, where $d' \leq d$. Every sub-dataset, along with a randomly selected subset of attributes, is used to train a decision tree. There are thus in total n decision trees that are trained independently with different sub-datasets and different attributes. Finally, either majority voting or averaging is applied to the n decision trees to get the final output of the RF, as shown in Figure 2.

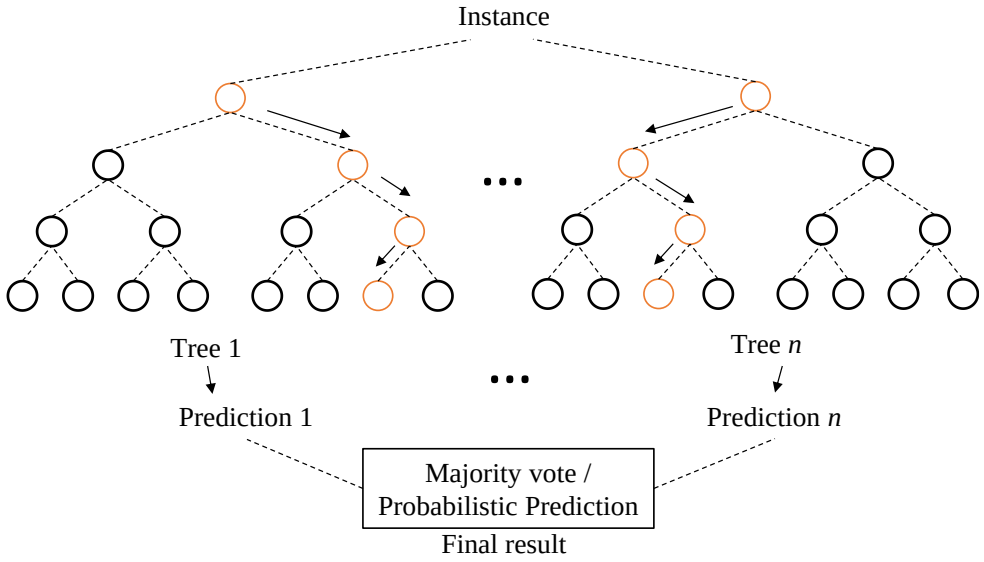


Figure 2. Illustration of a random forest (RF) model.

In general, an RF model can be used to classify an input instance into one of r classes $c_1, \ldots, c_r$. The procedure used to construct the RF model with n decision trees for a dataset of d data with k attributes has the following three major steps:

Step 1. Produce n sub-datasets from the original dataset of d data. Each sub-dataset is produced by drawing d' out of the d data with replacement, where $d' \leq d$.

Step 2. For each of the n sub-datasets, grow a decision tree by choosing the best splitting of internal tree nodes with the largest information gain for arbitrary k' attributes, $k' < k$. There are thus in total n decision trees to generate n classifications, each of which is one of r classes (i.e., labels) $c_1, \ldots, c_r$.

Step 3. Aggregate the results of the n trees to output the dominant class $c_{out} = \text{argmax}_{i=1}^{r} freq(c_i)$ as the final classification, where $freq(c_i)$ is the frequency that c_i appears among the n classifications. Note that the output may be adjusted to be with probabilistic classification, i.e., to output the classification frequencies (or probabilities) $freq(c_1), \ldots, freq(c_r)$ for all classes $c_1, \ldots, c_r$.

2.3. The Proposed AE-PRF Method

The proposed AE-PRF method first partitions the whole dataset as the training data, the validation data, and the test data. Figure 3 shows the processes of the proposed AE-PRF method. As shown in Figure 3, the data first undergo some preprocessing, and AE-PRF then applies the training data and the validation data to train an AE model. The AE model training is achieved by adjusting AE model weights properly with well-known error backpropagation and gradient descent mechanisms. The AE model can be used to reduce the data dimensionality and extract features from data as codes. Afterward, the codes of the training data are used to train an RF model to classify data into fraudulent data or normal data with associate classification probabilities. Moreover, the codes of the validation data are fed into the trained RF model to determine a proper threshold of classification probability to classify data with the best performance. Finally, for the verification purpose, the trained AE and RF models, along with the determined threshold can be applied to every test datum to check if it is fraudulent or normal.

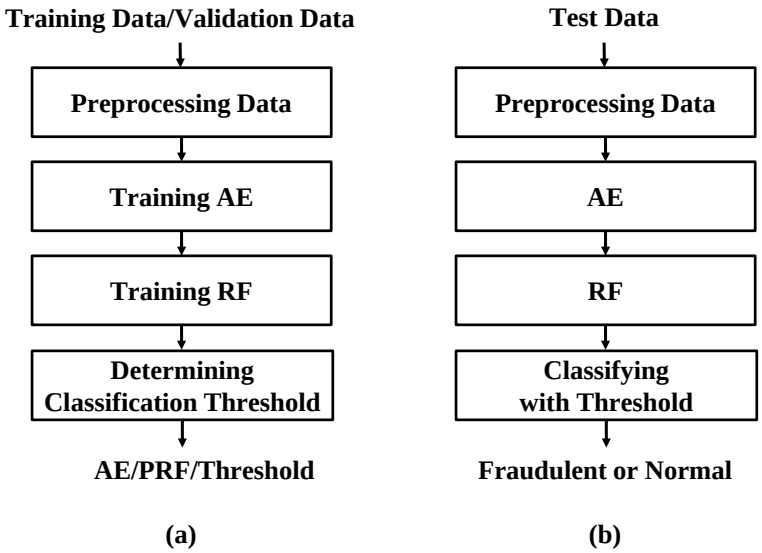


Figure 3. The illustration of AE-PRF: (**a**) the training process, and (**b**) the verification process.

Instead of directly using the RF model with a sole classification result, the AE-PRF method uses the RF model with probabilistic classifications to classify data. Specifically, the RF model with probabilistic classifications is used to classify a datum as fraudulent with probability p, and as normal with probability $1 - p$, where $0 \leq p \leq 1$. Afterward, AE-PRF outputs the final classification as fraudulent if p is larger than a pre-determined classification probability threshold θ. Different threshold values make AE-PRF generate different classification results. It is obvious that smaller θ values lead to a higher likelihood of classifying data as fraudulent. Fine-tuning the probability threshold θ value can provide AE-PRF with a customized classification result.

After the whole dataset is partitioned into the training data, the validation data, and the test data, the training process of AE-PRF can be started, as summarized in the following steps:

Step 1. Employ the training data to train the AE model AET and obtain the set T of training data feature codes.
Step 2. Train the RF model RFT with the set T of training data feature codes.
Step 3. Apply AET to the validation data to extract the set V of validation data feature codes.
Step 4. For threshold θ = 0 to 1 step s (=0.01), execute the following: Feed every code in V into RFT to output a probability p of fraud classification. If $p >$ θ, then the classification result is positive (fraudulent); otherwise, the classification result is negative (normal).
Step 5. Employ the classification results of all codes in V to find the threshold value θ* producing the best classification performance in terms of a specific metric M.

After the above-mentioned AE-PRF training process is finished, the verification process can be started, as summarized in the following steps.

Step 1. Apply AET to every test datum d to extract its feature code c.
Step 2. Feed the code c into RFT with the threshold value θ* to produce the classification result of d.

The pseudocode the proposed AE-PRF method is shown as Algorithm 1 below. The source code of AE-PRF implementation can be found at https://github.com/LinTzuHsuan/AE-PRF (accessed on 10 October 2021).

Algorithm 1 AE-PRF

Input: training data D_{train}, validation data $D_{validation}$, test data D_{test}, and metric M
Output: the classification result of each test datum (0 for normal or 1 for fraudulent)
Train the AE model AE_T with D_{train}
$T \leftarrow AE_T(D_{train})$
Train the RF model RF_T with T
$V \leftarrow AE_T(D_{validation})$
for $\theta \leftarrow 0$ to 1 step 0.01 **do**
    for each v in V **do**
        $p \leftarrow RF_T(v)$
        if $p > \theta$ **then** *result* $[\theta][v] \leftarrow 1$
        else *result* $[\theta][v] \leftarrow 0$
Find the best θ^* by comparing all *result* values in terms of metric M
$C \leftarrow AE_T(D_{test})$
for each c in C **do**
    $q \leftarrow RF_T(c)$
    if $q > \theta^*$ **then** *output* $[c] \leftarrow 1$
    else *output* $[c] \leftarrow 0$
return *output*

3. Related Work

The methods proposed in [12–15] are most related to AE-PRF. They all use the CCFD dataset for performance evaluation. None of them undergo data resampling except the methods proposed in [12]. Below, the related methods are elaborated one by one.

Three credit card fraud detection methods, namely naïve Bayes (NB), *k*-nearest neighbor (*k*-NN), and logistic regression (LR), are proposed in [12]. The best classification result is achieved by the *k*-NN method with $k = 3$. The *k*-NN method is a non-parametric supervised machine learning algorithm that can be used for classification and regression [26]. A test datum is classified into the dominant class of its *k* nearest neighbors' classes. Note that the random data resampling mechanism is adopted in [12] to address the data imbalance problem. Fraudulent data are oversampled and normal data are undersampled to make the ratio of fraudulent data to normal data 10:90 or 34:66 (≈1:2). The performance evaluation results show that data resampling can improve the performance of the *k*-NN method. However, it will be shown in this paper that data resampling does not necessarily improve the classification performance of the *k*-NN method.

Two unsupervised machine learning methods based on the AE model and the restricted Boltzmann machine (RBM) model are proposed in [13] for detecting credit card frauds. Like AE, RBM [27] can be sued to reconstruct input data. Both methods are unsupervised, as they need no data labels for training models. RBM can be regarded as a two-layer neural network with an input layer (visible) and a hidden layer. It is able to learn the probability distribution of the input data and thus can learn to reconstruct the data. This is achieved by fine-tuning the neural connection weights and biases through the processes of gradient descent and error back-propagation. For a new datum, either the trained AE or the trained RBM can be used to reconstruct the datum. The datum is assumed to be fraudulent if it has a large reconstruction error. As shown in [13], both AE and RBM have good fraud detection performance. However, AE is shown to have a better performance than RBM.

An unsupervised AE-based clustering method is proposed in [14] for detecting credit card frauds. The method uses an AE autoencoder with three hidden layers in both the encoder and the decoder. Moreover, it chooses the exponential linear unit (ELU) and the rectified linear unit (ReLU) as the activation functions of neurons in different layers. It also takes root mean square propagation (RMSProp) as the optimizer to yield the best result after performing several experiments. As shown in [14], the AE-base clustering method can achieve good classification performance by choosing an appropriate threshold of AE reconstruction errors to separate fraudulent data from normal data properly.

Twelve machine learning models for credit card fraud detection are studied in [15], including support vector machine (SVM) [28], naïve Bayes (NB), and feed-forward neural network (NN), etc. Furthermore, two ensemble learning mechanisms, namely adaptive boosting (AdaBoost) [29] and majority voting (MV), are combined with the twelve models to boost performance. Through comprehensive performance comparisons, SVM combined with AdaBoost (denoted as SVM + AdaBoost), and NN and NB combined with MV (denoted as NN + NB + MV) have comparably high performance. The SVM model generates a decision boundary in an increased or infinite-dimensional space, which is suitable for non-linear classification problems [30]. The AdaBoost method is an iterative method that adds a new weak classifier (i.e., classification model) in each iteration until all data are correctly classified, or the maximum iteration level has been reached. The NN + NB + MV model uses the feed-forward neural network and naïve Bayes concept [31] to perform fraud detection. The NN is an artificial neural network widely used in binary classification problems [32]. The NB is widely used for classification based on Bayes' theorem with strong independence assumptions between features [33]. It is good for the cases that the independence assumption fits. Due to MV, the NN + NB + MV model yields good classification results even when data are added with 10% to 30% of noise.

4. Performance Evaluation and Comparisons

4.1. Dataset and Data Resampling

The CCFD dataset [22] contains data generated by European cardholders within 2 days in September 2013. It has a total of 284,807 transactions, among which 492 are fraudulent. The dataset is highly imbalanced because fraudulent data account for 0.172% of total data. Each data entry has 31 attributes, including the transaction timestamp, the transaction amount, and the transaction class or label, which is 1 if the transaction is fraudulent, and 0, otherwise. It also has 28 principal component analysis (PCA) transformation values of transaction data. The PCA values are transformed from transaction data. They are for the purpose of hiding information like the cardholder identity and personal privacy data. Note that PCA is a feature extraction mechanism to project high-dimensional data into low-dimensional data without losing crucial information. It can also be used to transform data for the purpose of data dimensionality reduction, data feature extraction, and data de-identification.

As mentioned earlier, in order to make the CCFD dataset more balanced, data resampling schemes such as SMOTE [23], ADASYN [24], and T-Link [25] are applied to CCFD data before they are fed into AE-PRF. The three schemes are used to balance the numbers of majority class samples (or majority samples, for short) and minority class samples (or minority samples, for short). Their basic ideas are described below.

SMOTE is an oversampling technique. For a minority sample x_i, SMOTE first finds k nearest minority samples based on the k-NN scheme. It then selects a sample x_j out of the k nearest minority samples and generates a new minority sample x_{new} according to the equation: $x_{new} = x_i + \delta(x_j - x_i)$, where $\delta \epsilon [0, 1]$. The process to generate new minority samples continues until the number of newly generated minority samples reaches the pre-specified value.

ADASYN is also an oversampling technique. It is similar to SMOTE, but it adaptively generates new minority samples for a minority sample according to its imbalance degree. Specifically, for a minority sample x_i, its k nearest samples are first derived and its imbalance degree is defined as Δ_i/k, where Δ_i is the number of majority samples out of the k nearest samples of x_i.

T-Link is an undersampling technique. It tries to find a pair of a minority sample x_i and a majority sample x_j such that there is no sample x_k satisfying $d(x_k, x_j) < d(x_i, x_j)$ or $d(x_i, x_k) < d(x_i, x_j)$, where $d(u, v)$ is the Euclidean distance between samples u and v. It then removes the majority sample of every such pair so that the boundary between the majority class and the minority class is clearer and hence samples are easier to be classified.

4.2. Performance Metrics

The performance evaluation metrics, accuracy (ACC), the true positive rate (TPR), the true negative rate (TNR), and the false positive rate (FPR) are defined below in Equations (4)–(7), respectively.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{7}$$

In Equations (4)–(7), TP, FP, TN, and FN stand for the numbers of true positive, false positive, true negative, and false negative classifications (or predictions), respectively. A positive prediction is the one classifying a transaction as fraudulent, whereas a negative prediction is the one classifying a transaction as normal (i.e., not fraudulent). TP (respectively, FP) is the number of positive predictions for fraudulent (respectively, normal) transactions. TN (respectively, FN) is the number of negative predictions for normal (respectively, fraudulent) transactions. Note that TPR is also called sensitivity or recall, TNR is also called specificity, and FPR is also called the false alarm rate.

The area under the receiver operating characteristic curve (AUC) is a metric related to the receiver operating characteristic (ROC) curve. The ROC curve can be used as a tool to consider the tradeoff between TPR and FPR for a classifier based on threshold values. Different threshold values lead to different TPRs and FPRs. The ROC curve can be plotted by setting the x-axis as FPR and the y-axis as TPR, and the area under the ROC curve is then AUC. Larger AUC values correspond to better classifiers. If AUC has a value of 0.5, then the classifier is a no-skill classifier. If AUC has a value of 1, then the classifier is perfect.

The Matthews correlation coefficient (MCC) [34], as defined in Equation (8), can be regarded as a comprehensive metric, since it addresses TP, FP, TN, and FN at the same time. MCC has values within the range between -1 and $+1$, where the value of $+1$ indicates perfect predictions and -1 means entirely conflicting predictions. As stated in [35], MCC is suitable for both balanced and imbalanced datasets. Therefore, in the following performance evaluation, we consider MCC as an important metric.

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{8}$$

Several evaluation metrics, including ACC, TPR, TNR, MCC, AUC, recall, F1-score, log loss/binary cross-entropy, and categorical cross-entropy, can be used for evaluating the performance of classification methods. Among them, ACC, TPR, TNR, and MCC, which are defined in Equations (4)–(6) and (8), as well as AUC are commonly used for evaluating the performance of credit card fraud detection methods [12–15]. Specifically, the two most important metrics are TPR and MCC. The reason for the first metric, TPR, to be adopted in fraud detection is that the higher the TPR, the more fraudulent data can be detected, which is the main purpose of fraud detection. However, when evaluating the averaging performance of a model, MCC would be considered, because it takes every parameter including TP, TN, FP, and FN into consideration. To sum up, ACC, TPR, TNR, MCC, and AUC are deployed as metrics for performance evaluations and comparisons.

4.3. Performance Evaluation of AF-PRF

To evaluate the performance of the proposed AE-PRF method, the CCFD dataset is first partitioned into a training dataset of 64% data, a validation dataset of 16% data, and a test dataset of 20% data. All data undergo pre-processing such as the logarithmic transform on the amount of transaction and the second-to-day transform on the number

of seconds elapsed between the transaction and the first transaction in the dataset. The hyper-parameters of AE-PRF are described as follows. The AE model has five hidden layers with 26, 20, 18, 20, and 26 neurons, respectively, using the rectified linear unit (ReLU) as the activation function. The AE model uses the Adam as the optimizer, except for the first layer, using hyperbolic tangent (Tanh) instead. In order to prevent overfitting, L1 Regularization is applied to the first layer of the encoder. At the training stage, early stopping is adopted to prevent overfitting, using validation loss as the monitor. The dimension of the CCFD dataset data is reduced from 26 attributes to 18 by the trained AE. Well-defined feature extraction and dimensionality reduction algorithms (e.g., the AE model) make the detection/classification process more effective and efficient. The most important and influential features of the data will be focused on after dimensionality reduction.

The RF model of AE-PRF has 100 decision trees (estimators) and uses Gini impurity as the criterion, as defined in Equation (3). It generates probabilistic classification, i.e., it classifies the test datum as fraudulent with probability p, $0 \leq p \leq 1$.

The AE-PRF performance evaluation has two parts. The first part does not apply resampling mechanisms to data, whereas the second part applies resampling mechanisms to data. Below, we first describe the first part.

As mentioned earlier, AE-PRF uses the RF model with probabilistic classification with probability p to check if a test datum is classified as fraudulent. If p is larger than a pre-specified classification threshold θ, then the test datum is assumed to be fraudulent. Certainly, different threshold values lead to different classification performances. In order to find the best threshold confronting different requirements, it is necessary to fine-tune and shift the threshold and find the one which produces the best result in terms of specific metrics. More specifically, fine-tuning the threshold by testing different threshold values 0, 0.01, 0.02, . . . , 1 in agreement with the evaluation metric is the way to find the best threshold.

The threshold is first obtained by the ROC curve. To be precise, 101 different threshold θ values are applied to the AE-PRF classifier, ranging from 0 to 1 with the step interval of 0.01. Experiments are conducted 50 times to derive the average TPR and FPR, which in turn are used to plot the ROC curve, as shown in Figure 4. The zoomed-in version of Figure 4 is also given in Figure 5. We randomly repartitioned the dataset into a training set, a validation set, and a test set, and repeat the experiment 50 times to reduce biases of experimental results. The diagonal line in Figure 4 indicates the curve for a no-skill classifier. The upper left point on the ROC curve in Figure 4 indicates a model with perfect skill, which is computed by the geometric mean (or g-mean) of TPR and FPR (i.e., $\sqrt{\text{TPR} \times (1 - \text{FPR})}$. The g-mean of TPR and FPR is a good indicator of classification for imbalanced data. When it is optimized, a balance between the sensitivity (i.e., TPR) and specificity (i.e., TNR) is reached. The threshold recommended by the ROC curve is 0.03, which is the one corresponding to the best g-mean. The AUC of the ROC curve is 0.962, which is better than 0.960 of the method proposed in [13] and 0.961 of the method proposed in [14].

Similarly, the threshold is then tuned to obtain the best ACC, TPR, TNR, and MCC, as demonstrated in Figure 6. The ACC is very high with 101 thresholds, all about 0.99, except for $\theta = 0$. However, a high ACC alone cannot be interpreted as this fraud detection classifier being good enough. When dealing with highly imbalanced datasets, it is common to get a high ACC [36]. Therefore, other evaluation metrics must be taken into consideration. As for TNR, it also gets high scores with most of the thresholds, and the highest score is around 0.9998 achieved by $\theta = 0.25$. However, for the same reason as the ACC metric, because datasets of fraud detection problems are usually highly imbalanced, it tends to obtain a much higher TN value than FP value, which easily results in a high TNR [37].

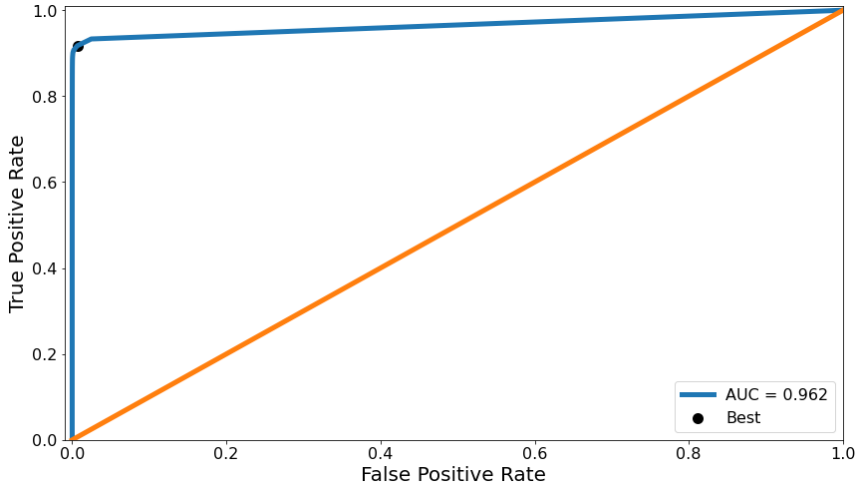


Figure 4. The ROC curve of AE-PRF.

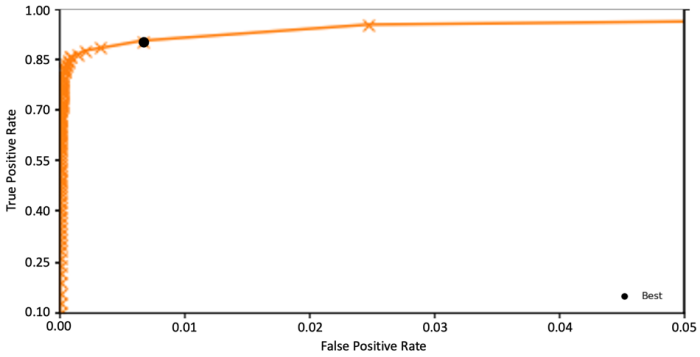


Figure 5. The zoomed-in ROC curve of AE-PRF (for FPR in [0.00, 0.05] and TPR in [0.01, 1.00]).

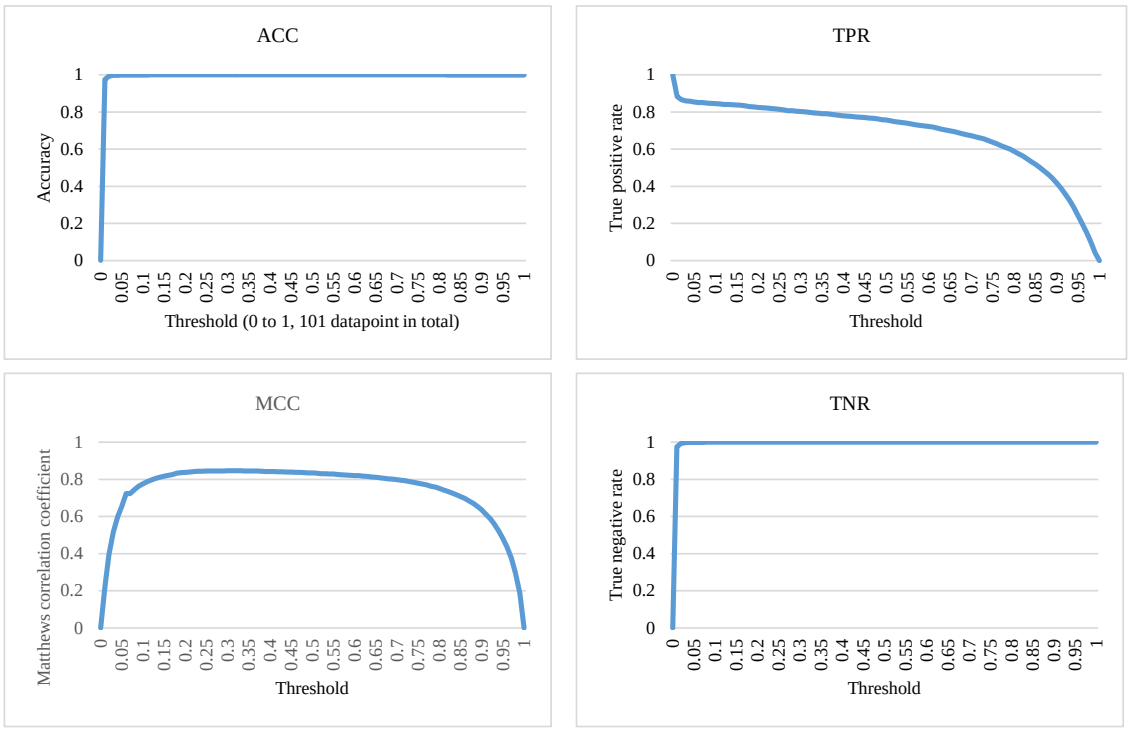


Figure 6. The ACC, TPR, TNR, and MCC of AE-PRF for different threshold values.

Therefore, this study considers MCC as a very important metric. Specifically, when considering and choosing the optimal threshold, this study just focuses on the thresholds generating MCC scores that are greater than 0.8. Thus, only threshold values ranging from 0.13 to 0.69 are considered. Consequently, the best average MCC is around 0.8456, obtained by setting $\theta = 0.25$.

However, it is risky to consider only one metric. In the credit card fraud detection problem, it is desirable to achieve higher TPRs so that more fraudulent data can be detected [36]. Nonetheless, if there are no restrictions, the highest TPR will always be achieved by $\theta = 0$. Under such a setting, not only ACC and TNR but also MCC will be very low, being 0. Thus, here, the premise is to set MCC greater than 0.5 and find the best TPR. In other words, if we want to detect as many fraudulent transactions as possible while maintaining a decent overall performance, another threshold must be adopted. To the best of our knowledge, the TPR of 0.89109 achieved by setting θ as 0.03 is the highest one ever seen while keeping MCC greater than 0.5. Note that setting θ as 0.03 is also recommended by the ROC curve using the g-mean of TPR and FPR.

Table 1 shows the details of performance metrics of AE-PRF for $\theta = 0.25$, which yields the best average MCC in our experiments. The details are the average score, the lowest scores (Minimum), the first quartile (Q1), the second quartile (Q2), the third quartile (Q3), and the highest scores (Maximum) of ACC, TPR, TNR, and MCC in 50 experiments with $\theta = 0.25$. The average score will be compared with those of other related methods later.

Table 1. Performance of AE-PRF in 50 times of experiments ($\theta = 0.25$).

Metrics	Average	Minimum	Q1	Q2	Q3	Maximum
ACC	0.99949	0.999410129	0.999455774	0.999494396	0.999522485	0.999578664
TPR	0.8142	0.75	0.808333333	0.816666667	0.825	0.84166667
TNR	0.9998	0.999704567	0.999788976	0.999803044	0.999831181	0.999887454
MCC	0.8441	0.811201026	0.834295395	0.845629405	0.853475139	0.870180134

Now, the second part of the AE-PRF performance evaluation is described. In this part, ADASYN itself and the combination of SMOTE and T-Link, denoted as SMOTE + T-Link, were applied to the training dataset for resampling data to make them more balanced. This was to verify whether the use of data resampling techniques can improve AE-PRF performance. However, if data resampling does not noticeably improve AE-PRF performance, then AE-PRF is said to be naturally suitable for dealing with imbalanced data.

The performance evaluation results of the AE-PRF without data resampling and with ADASYN and SMOTE + T-Link data resampling are shown in Table 2. The sampling strategies of ADASYN and SMOTE + T-Link are the same. That is, the ratio of the minority sample quantity over the majority sample quantity is set to 34:66 ($\approx$1:2) for both ADASYN and SMOTE + T-Link. Specifically, both ADASYN and SMOTE + T-Link adjust the number of fraudulent transactions to be 142,172, and the number of normal transactions to be 284,315 for the training dataset. As observed from Table 2, the performance of AE-PRF is not noticeably improved by data resampling. Moreover, AE-PRF using no data resampling even has better performance than AE-PRF using data resampling in terms of some metrics. It thus may be proper to say that AE-PRF is naturally suitable for dealing with imbalanced data.

Table 2. Performance of AE-PRF with and without data resampling.

Models	ACC	TPR	TNR	MCC
AE-PRF (θ = 0.03)	0.9973	0.8910	0.9975	0.5921
AE-PRF (θ = 0.25)	0.9995	0.8142	0.9998	0.8441
ADASYN AE-PRF (θ = 0.13)	0.9960	0.8613	0.9963	0.5018
ADASYN AE-PRF (θ = 0.57)	0.9995	0.8316	0.9998	0.8665
SMOTE + T-Link AE-PRF (θ = 0.11)	0.9965	0.8583	0.9967	0.5133
SMOTE + T-Link AE-PRF (θ = 0.51)	0.9995	0.8333	0.9998	0.8585

4.4. Perforamance Comparisons

Here, after several experiments, θ is set to be of values 0.25 and 0.03 for comparing AE-PRF and five related methods in terms of various performance metrics to demonstrate the superiority of AE-PRF. The five methods are the *k*-NN [12], AE [13], AE based clustering [14], SVM + AdaBoost [15], and NN + NB with MV [15]. All methods for comparison, including the proposed AE-PRF, use no data sampling. The performance comparisons were performed in terms of ACC, TPR, TNR, AUC, and MCC.

Table 3 shows the performance comparison results of AE-PRF and other five related methods. The highest scores in Table 3 are in boldface. It can be seen that AE-PRF outperformed others in almost all metrics. As for AE-PRF with θ = 0.25, it had the highest ACC of 0.9995, the highest TNR of 0.9998, and the highest MCC of 0.8441. However, its TPR of 0.8142 was lower than the highest score of 0.8835 achieved by *k*-NN [12]. Therefore, another threshold was adopted, AE-PRF with θ = 0.03 had the highest TPR of 0.89109 and comparable high ACC, TNR, and MCC. If the main goal of the credit card fraud detection is to achieve as high TPR as possible while maintaining a decent MCC (say $\geq$ 0.5), then AE-PRF with θ = 0.03 is the best one to choose.

Table 3. Performance comparisons of AE-PRF and related methods.

Research	Methods	ACC	TPR	TNR	MCC	AUC
Awoyemi et al. [12]	*k*-NN	0.9691	0.8835	0.9711	0.5903	-
Pumsirirat et al. [13]	AE	0.97054	0.83673	0.97077	0.1942	0.9603
Zamini et al. [14]	AE-based clustering	0.98902	0.81632	0.98932	0.3058	0.961
Randhawa et al. [15]	SVM with AdaBoost	0.99927	0.82317	0.99957	0.796	-
Randhawa et al. [15]	NN+NB with MV	0.99941	0.78862	0.99978	0.823	-
This Research	AE + PRF (θ = 0.03)	0.99738	**0.89109**	0.99757	0.5921	**0.962**
This Research	AE + PRF (θ = 0.25)	**0.9995**	0.8142	**0.9998**	**0.8441**	**0.962**

Note that the *k*-NN method proposed in [12] has two versions, one using data resampling and the other using no data resampling. However, only the version using no data resampling is compared with the proposed AE-PRF method in Table 3. This is because when we re-implement the *k*-NN method and apply random data resampling to the re-implemented *k*-NN method, the performance of the re-implemented *k*-NN does not conform with the performance results shown in [12]. As demonstrated in Table 4, the data resampling even makes *k*-NN have bad performance. The research [12] likely applied data resampling to the whole data, including the training and the test data, whereas we apply data resampling to only the training data. We confirm this by applying random data resampling to the whole data and then running the re-implemented *k*-NN method. As observed from Table 4, if the whole data is resampled, then the performance results of the original *k*-NN and the re-implemented *k*-NN are quite similar. However, not all

test data can be obtained in advance and each test datum should be classified separately. Resampling all data, including training data and test data, seems to be impractical.

Table 4. Performance of *k*-NN [12] and re-implemented *k*-NN with and without data resampling.

Methods	ACC	TPR	TNR	MCC
k-NN [12] (without resampling)	0.9691	0.8835	0.9711	0.5903
k-NN [12] (with all data 34:66 resampling)	0.9792	0.9375	1.0	0.9535
Re-implemented *k*-NN (without resampling)	0.9977	0.7483	0.9981	0.5512
Re-implemented *k*-NN (with only training data resampling)	0.9817	0.1881	0.9832	0.0556
Re-implemented *k*-NN (with all data 34:66 resampling)	0.9832	0.9494	1.0	0.9624

5. Conclusions

This paper proposes a fraud detection method called AE-PRF. It employs AE to reduce data dimensionality and extract data features. Moreover, it utilizes RF with probabilistic classification to classify data as fraudulent along with an associated probability. AE-PRF outputs the final classification as fraudulent if the associated probability exceeds a predetermined probability threshold θ.

The CCFD dataset [22] was applied to evaluate the performance of AE-PRF. Since the CCFD dataset is highly imbalanced, data resampling schemes like SMOTE [23], ADASYN [24], and T-Link [25] were applied to the CCFD dataset to balance the numbers of normal and fraudulent transactions. Experimental results showed that the performance of AE-PRF does not vary much whether resampling schemes are applied to the dataset or not. This indicates that AE-PRF is naturally suitable for handling imbalanced datasets without data resampling.

The performance evaluation results of AE-PRF without data resampling were compared with those of related methods such as *k*-NN [12], AE [13], AE-based clustering [14], SVM with AdaBoost [15], and NN + NB with MV [15]. The comparison results show that AE-PRF with $\theta = 0.25$ has the highest ACC, TNR, MCC, and AUC, and has comparably high TPR. As for AE-PRF with $\theta = 0.03$, it has the highest TPR and AUC, and comparable high ACC, TNR, and MCC. The CCFD dataset is partitioned into a training dataset of 64% data, a validation dataset of 16% data, and a test dataset of 20% data for evaluating AE-PRF performance. We tried another extreme partition, a training dataset of 40% data, a validation dataset of 10% data, and a test dataset of 50%, which does not yield a good result because of the insufficient training data.

It is more persuasive to compare AE-PRF to existing methods using the same dataset for performance evaluation. Since the CCFD dataset was adopted by many existing methods and it is the most detailed public dataset, this paper adopted the CCFD dataset for performance evaluation and comparison. However, in order to test the robustness and effectiveness of AE-PRF, we need to adopt some other datasets, especially private datasets, because there are few public datasets for credit card fraud detection due to privacy issues. In the future, we plan to cooperate with credit card issuers and/or banks to obtain datasets for verifying the robustness and the effectiveness of AE-PRF.

In the future, we will try to improve AE-PRF performance by fine-tuning the hyperparameters of the AE and the RF models. We will also try to apply AE-PRF to a variety of applications for evaluating AE-PRF's applicability. Furthermore, we will investigate the explainability of AE-PRF and try to enhance AE-PRF's explainability by leveraging novel explainable AI (XAI) schemes proposed in [38–40] for AE and RF.

Author Contributions: Conceptualization, T.-H.L. and J.-R.J.; funding acquisition, J.-R.J.; investigation, T.-H.L. and J.-R.J.; methodology, T.-H.L. and J.-R.J.; software, T.-H.L.; supervision, J.-R.J.; validation, T.-H.L. and J.-R.J.; writing—original draft, T.-H.L. and J.-R.J.; writing—review & editing, T.-H.L. and J.-R.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology (MOST), Taiwan, under the grant number 109-2622-E-008-028-.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. de Best, R. Credit Card and Debit Card Number in the U.S. 2012–2018. Statista. 2020. Available online: https://www.statista.com/statistics/245385/number-of-credit-cards-by-credit-card-type-in-the-united-states/#statisticContainer (accessed on 10 October 2021).
2. Voican, O. Credit Card Fraud Detection using Deep Learning Techniques. *Inform. Econ.* **2021**, *25*, 70–85. [CrossRef]
3. The Nilson Report. Available online: https://nilsonreport.com/mention/1313/1link/ (accessed on 20 December 2020).
4. Taha, A.A.; Sharaf, J.M. An intelligent approach to credit card fraud detection using an opti-mized light gradient boosting machine. *IEEE Access* **2020**, *8*, 25579–25587. [CrossRef]
5. Dal Pozzolo, A. Adaptive Machine Learning for Credit Card Fraud Detection. Ph.D. Thesis, Université Libre de Bruxelles, Brussels, Belgium, 2015.
6. Lucas, Y.; Portier, P.-E.; Laporte, L.; Calabretto, S.; Caelen, O.; He-Guelton, L.; Granitzer, M. Multiple perspectives HMM-based feature engineering for credit card fraud detection. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, ACM, New York, NY, USA, 8–12 April 2019; pp. 1359–1361.
7. Wiese, B.; Omlin, C. Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks. In *Studies in Computational Intelligence*; Springer Science and Business Media LLC: Berlin, Germany, 2009; pp. 231–268.
8. Jurgovsky, J.; Granitzer, M.; Ziegler, K.; Calabretto, S.; Portier, P.-E.; He-Guelton, L.; Caelen, O. Sequence classification for credit-card fraud detection. *Expert Syst. Appl.* **2018**, *100*, 234–245. [CrossRef]
9. Zhang, F.; Liu, G.; Li, Z.; Yan, C.; Jiang, C. GMM-based Undersampling and Its Application for Credit Card Fraud Detection. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
10. Ahammad, J.; Hossain, N.; Alam, M.S. Credit Card Fraud Detection using Data Pre-processing on Imbalanced Data—Both Oversampling and Undersampling. In Proceedings of the International Conference on Computing Advancements, New York, NY, USA, 10–12 January 2020; ACM Press: New York, NY, USA, 2020.
11. Lee, Y.-J.; Yeh, Y.-R.; Wang, Y.-C.F. Anomaly Detection via Online Oversampling Principal Component Analysis. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 1460–1470. [CrossRef]
12. Awoyemi, J.O.; Adetunmbi, A.O.; Oluwadare, S.A. Credit card fraud detection using machine learning techniques: A comparative analysis. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 29–31 October 2017; pp. 1–9.
13. Pumsirirat, A.; Yan, L. Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 18–25. [CrossRef]
14. Zamini, M.; Montazer, G. Credit Card Fraud Detection using autoencoder based clustering. In Proceedings of the 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 17–19 December 2018; pp. 486–491. [CrossRef]
15. Randhawa, K.; Loo, C.K.; Seera, M.; Lim, C.P.; Nandi, A.K. Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access* **2018**, *6*, 14277–14284. [CrossRef]
16. Lucas, Y.; Johannes, J. Credit card fraud detection using machine learning: A survey. *arXiv* **2020**, arXiv:2010.06479.
17. Nikita, S.; Pratikesh, M.; Rohit, S.M.; Rahul, S.; Chaman Kumar, K.M.; Shailendra, A. Credit card fraud detection techniques—A survey. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering, Vellore, India, 24–25 February 2020.
18. Rumelhart, D.E.; Geoffrey, E.H.; Ronald, J.W. Learning internal representations by error propagation. *Calif. Univ. San Diego La Jolla Inst. Cogn. Sci.* **1985**, *8*, 318–362.
19. Liaw, A.; Matthew, W. Classification and regression by random Forest. *R News* **2002**, *2*, 18–22.
20. Seeja, K.R.; Zareapoor, M. FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining. *Sci. World J.* **2014**, *2014*, 1–10. [CrossRef]
21. Zhang, C.; Gao, W.; Song, J.; Jiang, J. An imbalanced data classification algorithm of improved autoencoder neural network. In Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 95–99.
22. Credit Card Fraud Detection Dataset. Available online: https://www.kaggle.com/mlg-ulb/creditcardfraud (accessed on 20 August 2020).
23. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
24. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the IEEE International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
25. Tomek, I. Two Modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 769–772. [CrossRef]
26. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.

27. Sutskever, I.; Geoffrey, E.H.; Graham, W.T. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*; University of Toronto: Toronto, ON, Canada, 2009.
28. Bhattacharyya, S.; Jha, S.; Tharakunnel, K.; Westland, J.C. Data mining for credit card fraud: A comparative study. *Decis. Support Syst.* **2011**, *50*, 602–613. [CrossRef]
29. Margineantu, D.; Dietterich, T. Pruning Adaptive Boosting. In Proceedings of the 14th International Conference on Machine Learning, ICML, Guangzhou, China, 18–21 February 1997.
30. Naveen, P.; Diwan, B. Relative Analysis of ML Algorithm QDA, LR and SVM for Credit Card Fraud Detection Dataset. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 7–9 October 2020; pp. 976–981.
31. Rish, I. An Empirical Study of the Naive Bayes Classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. 2001, Volume 3. No. 22. Available online: https://www.cc.gatech.edu/fac/Charles.Isbell/classes/reading/papers/Rish.pdf (accessed on 20 August 2020).
32. Jeatrakul, P.; Wong, K.W. Comparing the performance of different neural networks for binary classification prob-lems. In Proceedings of the 2009 Eighth International Symposium on Natural Language Processing, Bangkok, Thailand, 20–22 October 2009.
33. Murphy, K.P. *Naive Bayes Classifiers*; University of British Columbia: Vancouver, BC, Canada, 2006.
34. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 1–13. [CrossRef] [PubMed]
35. Saito, T.; Rehmsmeier, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef] [PubMed]
36. Chawla, N.V. Data Mining for Imbalanced Datasets: An Overview. *Data Min. Knowl. Discov. Handb.* **2009**, 875–886. [CrossRef]
37. Lakshmi, T.J.; Prasad, C.S.R. A study on classifying imbalanced datasets. In Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), Guntur, India, 19–20 August 2014; pp. 141–145.
38. Assaf, R.; Giurgiu, I.; Pfefferle, J.; Monney, S.; Pozidis, H.; Schumann, A. An Anomaly Detection and Explainability Framework using Convolutional Autoencoders for Data Stor-age Systems. *IJCAI* **2020**, 5228–5230. [CrossRef]
39. Antwarg, L.; Miller, R.M.; Shapira, B.; Rokach, L. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Syst. Appl.* **2021**, *186*, 115736. [CrossRef]
40. Fernández, R.R.; de Diego, I.M.; Aceña, V.; Fernández-Isabel, A.; Moguerza, J.M. Random forest explainability using counterfactual sets. *Inf. Fusion* **2020**, *63*, 196–207. [CrossRef]

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 48 (2015) 679 – 685

Procedia Computer Science

International Conference on Intelligent Computing, Communication & Convergence (ICCC-2015)

Conference Organized by Interscience Institute of Management and Technology, Bhubaneswar, Odisha, India

Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier

Masoumeh Zareapoor[a], Pourya Shamsolmoali[a,b]

[a] *Department of Computer science, Jamia Hamdard University, New Delhi, India*
[b] *Department of Computer Science, *Baghin University, Kerman, Iran*

Abstract

Credit card fraud is increasing considerably with the development of modern technology and the global superhighways of communication. Credit card fraud costs consumers and the financial company billions of dollars annually, and fraudsters continuously try to find new rules and tactics to commit illegal actions. Thus, fraud detection systems have become essential for banks and financial institution, to minimize their losses. However, there is a lack of published literature on credit card fraud detection techniques, due to the unavailable credit card transactions dataset for researchers. The most commonly techniques used construct the fraud detection model. The performance evaluation is performed on real life credit card transactions dataset todemonstrate the benefit of the bagging ensemble algorithm.

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)

Peer-review under responsibility of scientific committee of International Conference on Computer, Communication and Convergence (ICCC 2015)
doi:10.1016/j.procs.2015.04.201

fraud detection methods are Naïve Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbor algorithms (KNN). These techniques can be used alone or in collaboration using ensemble or meta-learning techniques to build classifiers. But amongst all existing method, ensemble learning methods are identified as popular and common method, not because of its quite straightforward implementation, but also due to its exceptional predictive performance on practical problems. In this paper we trained various data mining techniques used in credit card fraud detection and evaluate each methodology based on certain design criteria. After several trial and comparisons; we introduced the bagging classifier based on decision three, as the best classifier to construct the fraud detection model. The performance evaluation is performed on real life credit card transactions dataset to demonstrate the benefit of the bagging ensemble algorithm.

*Keywords:*Credit card fraud; Fraud detection technique; Data mining; Bagging ensemble classifer.

1.Introduction

Credit card is considered as a "nice target of fraud" since in a very short time attackers can get lots of money without much risk and most of the time the fraud is discovered after few days. To commit the credit card fraud either offline or online, fraudsters are looking for sensitive information such as credit card number, bank account and social security numbers. In case of offline payment (which using credit card physically) to perform the fraudulent transactions, an attacker has to steal the credit card itself, while in the case of online payment (which occurs over internet or phone), the fraudsters should be steal costumer's identity. Credit card fraud is a significant issue and has considerable cost for banks and card issuer companies. Thus, with this massive problem in transaction system, banks take credit card fraud very seriously, and have highly sophisticated security systems to monitor transactions and detect the frauds as quickly as possible once it is committed. A secured and trusted banking payment system requires high speed verification and authentication mechanisms that let legitimate users easy conduct their business, while flagging and detecting suspicious transaction attempts by others. Fraud detection has become a vital activity in order to decrease the impact of fraudulent transactions on service delivery, costs, and reputation of the company. According to Kount, one of the top five fraud detection consultants revealed by *topcreditcardprocessors.com* in the month of August 2013[1], 40% of the total financial fraud is related to credit card and the loss of amount due to credit card fraud worldwide is $5.55 billion. There are various methods used for fraud detection each of them tries to increase the detection rate while keeping false alarm rate at minimum. Different methods have been used for fraud detection including such as, Bayesian algorithm [15][9], K-Nearest Neighbor[17][7], Support Vector Machine [17][6] etc. Statistical fraud detection methods have been divided into two broad categories [3][11]: supervised and unsupervised. In supervised fraud detection methods, models are estimated based on the samples of fraudulent and legitimate transactions, to classify new transactions as fraudulent or legitimate. While in unsupervised fraud detection, outliers or unusual transactions are identified as potential cases of fraudulent transactions. Both these fraud detection methods predict the probability of fraud in any given transaction. The objective of this paper is to perform a comprehensive review of various fraud detection methods and selects some innovative method and technique for discussion.

2. Challenges in credit card fraud detection

Credit card fraud detection is one of the most explored domains of fraud detection and relies on the automatic analysis of recorded transactions to detect fraudulent behaviour. Furthermore the problem of credit card fraud detection has many constraints. Here we discussed various challenges in credit card fraud detection are:

2.1 Non-availability of real data set

One of the biggest issues associated with credit card fraud detection is the unavailability of dataset that researchers can perform the research on as it mentioned by many authors [11][14][16]. The reason of unavailability of real world data is, banks and financial institution are not ready to reveal their sensitive customer transaction data due to privacy reasons.

2.2 Unbalanced Data Set

Credit card fraud datasets are highly skewed data (where many more of data is legitimate, and a few of

[1] www.prweb.com/releases/2013/8

them is fraudulent), and the legal and fraud transactions vary at least hundred times [14]. Generally, in real case 98% of the transactions are legal while only 2% of them are fraud [13].

2.3 Size of the Data Set

Chan et al point out in [5] that, millions of credit card transactions are processed every day. Analyzing such enormous amounts of transactions requires highly competent techniques that scale well, as well as requiring considerable computing power. It creates certain restrictions for the researchers.

2.4 Determining the appropriate evaluation parameters

There are two very common measures for the fraud detection techniques: false-positive and false-negative rates. These two measures have an opposite relationship, one decrease and other one increase. Accuracy is not a suitable metrics for credit card fraud detection technique since; the dataset is highly imbalanced [3]. Therefore with very high accuracy all fraudulent transactions can be misclassified. The error cost of misclassifying fraudulent instances is higher than the error cost of misclassifying legitimate instances, it is important to study not only the precision (correct classified instances) but also the sensibility (correct classified fraudulent instances) of each case.

2.5 Dynamic behaviour of fraudster

Fraudsters having dynamic behaviour mean that the fraudsters change their behaviour over the time to get through any new detection system and modify fraud styles. So, fraud is becoming increasingly more complex and sophisticated which which is not even predictable by human experts.

But with these challenges, credit card fraud detection is still a fashion and hard research topic.

3. Data mining techniques

We investigated the performance of five states of art techniques in predicting credit card fraud: Support Vector Machines (SVM), Naïve Bayes (NB), KNN and Bagging ensemble classifier. In the paragraphs below, we briefly describe the techniques employed in this study.

3.1 Naive Bayes classifiers

Naïve Bayes is a supervised machine learning method that uses a training dataset with known target classes to predict the class of future instances. This algorithm was first introduced by John and Langley (1995) [2]. In simplest terms, a Naïve Bayes method assumes that the "presence or absence" of a particular attribute of a set is not based on the presence or absence of any other attributes in the same set. Experiments on the real world dataset have shown that the NB algorithm performs comparably well. However, this technique is named by the name "Naïve" because it naively assumes independence of the attributes given the class [17][2]. Then the classification is done by applying "Bayes" rule to calculate the probability of the correct class which is the particular attributes of the credit card transaction. Bayes theory is calculated as:

$$\textit{Prior Probability of Z: } \frac{\text{Number of Z instances}}{\text{total number of instances}} \qquad \textit{Likelihood of Y given Z: } \frac{\text{Number of Z in vicity of Y}}{\text{total number of Z}}$$

In the Bayesian theory, the final classification is produced by combining both information (likelihood, priori), to form a *posterior probability* which is called Bayes rule.

$$\textit{Posterior} = (\text{Prior} * \text{Likelihood}) / (\text{Evidence})$$

It has a good performance with small amount of training data. It is used to solve both binary classification problem and multiclass classification problem.

3.2 K-Nearest Neighbor algorithm

The k-Nearest Neighbor (KNN) technique is a simplest technique that stores all available instances and then

classifies any new instances based on a similarity measure. KNN has been used in statistical estimation and pattern recognition in the beginning of 1970's. The KNN algorithm is an example of an instance based learner. In other word, all of the learning models are "instance based," as well, because they start with a set of instances as the initial training information. In the nearest-neighbour classification method, each new instance is compared with existing ones by using a distance metric, and the closest existing instance is used to assign the class to the new one. Sometimes more than one nearest neighbour is used, and the majority class of the closest *k* neighbours is assigned to the new instance. The concept of the instance-based nearest-neighbour algorithm was first introduced by Aha, Kibler, and Albert (1991) [4]. K- Nearest neighbour based credit card fraud detection techniques require a distance or similarity measure defined between two data instances [17][1]. In process of KNN, we classify any incoming transaction by calculating a nearest point to the new incoming transaction. Then if the nearest neighbour be fraudulent, then the transaction indicates as a fraud. The value of K is generally small and odd to break the ties (typically 1or 3). Larger K values can help to reduce the effect of noisy data set. In this algorithm, distance between two data instances can be calculated in different ways. For continuous attributes, Euclidean distance is a good choice. For categorical attributes, a simple matching coefficient is often used. The most important pitfall of KNN algorithm is that unrelated attributes have a large negative impact on the training process of the K-Nearest neighbour and because of these irrelevant attributes the training of classifiers based on these algorithm can often be inefficient and impractical [17].

3.3 Support Vector Machines (SVMs)

The Support Vector Machine (SVM) is statistical learning technique which is especially suitable for binary classification technique [14][6][17] such as credit card fraud detection techniques which only two classes are needed, namely the legitimate and fraudulent class. The goal of the SVM method is to construct a "hyperplane" which do separate the data instances into two classes:- positive and negative [8][30]. The strength of SVMs comes from two main properties:- kernel representation and margin optimization. Kernels, such as radial basis function (RBF) kernel, can be used to learn complex regions. A kernel function represents the dot product of projections of two data instance in a high dimensional feature space. The basic technique finds the smallest "hypersphere" in the kernel space that contains all training instances, and then determines on which side of "hypersphere" a test instance lies. This classifier finds the maximum margin hyper plane, and it classifies all training instances correctly by separating them into correct classes through a hyper plane. The maximum margin hyper plane is the one that gives the greatest separation between the classes. The instances that are nearest to the maximum margin hyper plane are called support vectors. In credit card fraud detection if a test instance lies within the learned region, it is stated as normal; else it is declared as anomalous. SVM methods require large training dataset sizes in order to achieve maximum prediction accuracy. However, regular SVM method is invalid to the imbalanced data sets. Because in imbalanced data sets, the learned boundary is close to the minority instances, so SVM should be biased in a way that will push the boundary away from the positive samples [17].

3.4 Bagging ensemble classifier based on decision tree

Bagging classifier is an ensemble technique which was proposed by Leo Breiman in 1994 [12]. It can be handle classification and regression methods. It is designed to improve the stability and accuracy of machine learning algorithms used in classification and regression. It works by combining classifications of randomly generated training sets to form a final prediction. Such techniques can typically be used as a variance reduction technique by randomization into its construction procedure and then creating an ensemble out of it. Bagging classifier has attracted much attention, due to its simple implementation and the improving accuracy. Thus, we can call bagging as a "smoothing operation" that has advantage when intending to improve the predictive performance of regression or classification trees. The basic principle behind of this ensemble method is that a group of "weak learners" can come together to form a "strong learner". Bagging grows many decision trees. Here each individual decision tree is a "weak learner", while all the decision trees taken together are a "strong learner". When a new instance has to be classified, it is done repeatedly to each of the trees in the ensemble. Each tree gives a "vote" for a class. The final prediction for the new instance's class is gained by the class having maximum votes. In this paper we used bagging classifier [12], with the decision tree algorithm J48 based on the C4.5 model as the single classifier to construct the ensemble. The reason for selecting decision three as a single classifier for our ensemble is that, our dataset is highly imbalanced, so decision three algorithm presents a very good behavior by weighting the results of the trees and reducing the variance of the dataset and the overfitting. Bagging ensemble classifier is fast and they can efficiently

handle unbalanced and large databases with thousands of features.

4. Experimental setup

The objective of this paper is to examine the evaluation performance of three advanced data mining techniques, with the well known and proposed bagging ensemble classifier, for credit card fraud detection technique. In this work we used 10 fold cross validation techniques.

4.1 Dataset

In the field of credit card fraud detection technique there are different types of datasets with different fraud properties, e.g. type of fraud, number of fraudulent records, variety of fraud, the distribution of fraudulent transactions among legal transactions. In this paper for evaluating the state of art techniques we used the real world credit card dataset which is obtained from UCSD-FICO competition. The competition was organized by FICO, the leading provider of analytics and decision management technology, and the University of California, San Diego (UCSD). The dataset is a real dataset of e-commerce transactions and the objective was to detect anomalous e-commerce transactions. The obtained dataset include of 100,000 records of credit card transactions. Each record has 20 fields. The data given to us was already labeled by bank, as legitimate and fraudulent. The ratio of legitimate transactions to fraudulent transactions approximately is 100:3. This means that among the 100,000 records, 2.8% (2293 records) are fraudulent transactions, and 97.2% (records) are legitimate transactions. The data are sampled from 98 days period. In our dataset we couldn't find any difference between the attributes in legitimate and fraudulent transactions, due to appearance of fraudulent behavior more and more like legitimate ones. Figure 1 present the credit card transactions dataset available in this paper as follow:

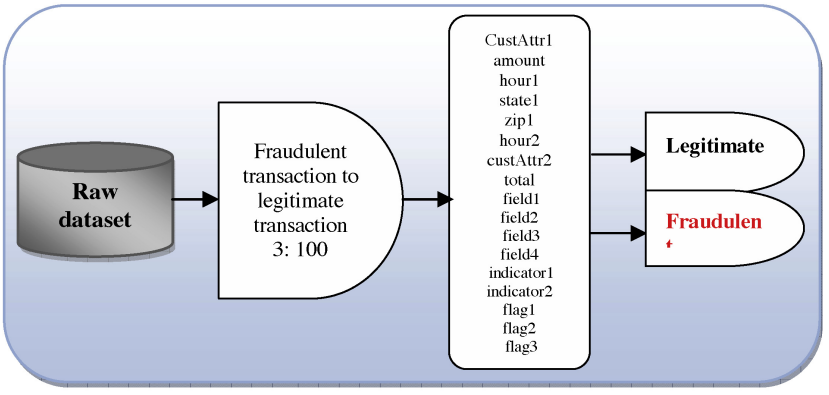


Fig. 1. Dataset description

4.2 Training and test data

As mentioned in introduction section, the given dataset in this paper is highly imbalanced data which is the nature of credit card transaction dataset. We divided our dataset into 4 groups. As shown in Table 1, the fraud rates in these modeling datasets Df1, Df2, Df3, Df4 is approximately 20%, 15%, 10%, 3% respectively.

5.1 Performance measures

In this paper we didn't consider some of common metrics like accuracy and error rate, since they are known to be bias metrics in the case of imbalanced dataset. For fraud detection domain, the "fraud catching rate" and "false alarm rate" are the criteria metrics. We are evaluated the performance of the various techniques in terms of 4 classification metrics relevant to credit card fraud detection [18] – Fraud Catching Rate, False Alarm Rate, Balanced Classification Rate and Matthews Correlation Coefficient. Here, fraud is considered as positive class and legal as negative class and hence the meaning of the terms TP, TN, FP and FN are defined as follows:

True Positive (TP) = Number of fraud transactions predicted as fraud

True Negative (TN) = Number of legal transactions predicted as legal

False Positive (FP) = Number of legal transactions predicted as fraud

False Negatives (FN) = Number of fraud transactions predicted as legal

6. Results

This section presents result of our evaluation performance model developed from the dataset. In this paper, we compare several standard classifiers with the bagging ensemble classifier which is novel technique in credit card fraud detection technique.

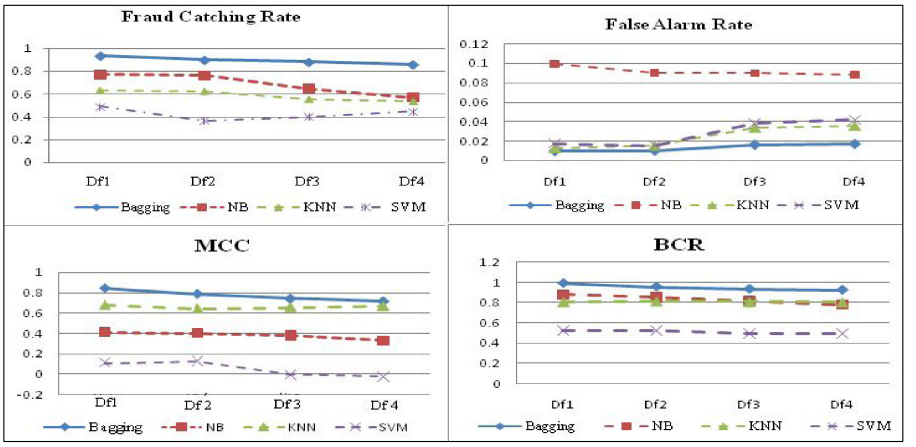


Fig. 2. Performance of all classifies in terms of MCC, BCR, FAR, FCR LS using real world credit card dataset

As shown in figure 2, bagging classifier based on decision three, very well detect the fraudulent transactions, the fraud catching rate is high keeping and the false alarm rate very low. While other methods have problem to detect the fraudulent transactions by increasing the number of false alarm rate. Also, the results show the superiority of the bagging classifier based decision three, irrespective to the rate of fraud. It means that, the bagging ensemble classifier has stable performance in during the training and testing evaluation, and is independent to rate of frauds. Even it shows the better result with highly imbalanced dataset (Df4). When we compare the methods in terms of false alarm rate, we observe that the behaviour of the bagging technique is similar in duration of evaluation (for Df1, Df2, Df3, and Df4). While the performance of other classifiers is lower than bagging classifier. Two balanced metrics-BCR & MCC are used to evaluate the capability of various techniques for handling class imbalance and figure 2, shown bagging classifier has a very good performance according to these measures compared with other classifiers. This is explained by the fact that bagging ensemble classifier is more suitable for credit card fraud detection, since the nature of dataset is highly imbalanced, and it has capability to handle the imbalanced dataset. But other standard classifiers are known to be bias classifiers.

7. Conclusion

This paper examined the performance of three states of art data mining techniques, with bagging ensemble classifier based on decision three algorithms which is a novel technique in area of credit card fraud detection system. A real life dataset on credit card transactions is used for our evaluation. And we found that, the bagging classifier based on decision three works well with this kind of data since it is independent of attribute values. The second feature of this novel technique in credit card fraud detection is its ability to handle class imbalance.

This is incorporated in the model by creating four sets of dataset (Df1, Df2, Df3, DF4) which the fraud rate in each of them were 20%, 15%, 10%, 3% respectively. Bagging classifier based decision three algorithm performance is found to be stable gradually during the evaluation. More over the bagging ensemble method takes very less time, which is also an important parameter of this real time application, because in fraud detection domain time is known one of the important parameter.

ACKNOWLEDGEMENTS

The authors wish to thank Dr Kohei Hayashi, Nara Institute of Science and Technology, Japan, and Dr.Haiqin Yang, Chinese University of Hong Kong for providing the Dataset.

Table1. Training and testing dataset

Group of dataset	Legal	Fraud	Total	Fraud rate
Df1	14170	2834	17004	**20%**
Df2	18895	2834	21729	**15%**
Df3	28340	2834	31174	**10%**
Df4	97166	2834	10000	**3%**

References

[1] S. Kotsiantis, D. Kanellopoulos, P. Pintelas (2006). Handling imbalanced datasets: A review. International Transactions on Computer Science and Engineering.
[2] G.H. John, P. Langley (1995). Estimating continuous distributions in Bayesian classifiers. in: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, (1995); 338 – 345.
[3] R.J. Bolton, D.J. Hand (2001). Unsupervised profiling methods for fraud detection. *In Conference on credit scoring and credit control*, Edinburgh.
[4] D. Kibler, D.W. Aha, M. Albert (1989). Instance-based prediction of real-valued attributes. Computational Intelligence, Vol(5); 51-57.
[5] P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo (1999). Distributed Data Mining in Credit Card Fraud Detection. *IEEE Intelligent Systems*, pp 67–74.
[6] C. Cortes, V. Vapnik (1995). Support vector networks. Machine Learning. , 20:273–297.
[7] T.M. Cover, P.E. Hart (1967). Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory*, 13(1):21–27.
[8] G. Potamitis (2013). Design and Implementation of a Fraud Detection Expert System using Ontology-Based Techniques. A dissertation submitted to the University of Manchester for the degree of Master of Science in the Faculty of Engineering and Physical Sciences.
[9] E. David (2012). Bayesian inference-the future of online fraud protection. *Computer Fraud & Security*, 8-11.
[10] S. Ghosh, D.L. Reilly. (1994). Credit Card Fraud Detection with a Neural- Network. *In Proceedings of the International Conference on System Science*, pages 621-630.
[11] Jha.Sanjeev, G. Montserrat, J.C.Westland (2012). Employing transaction aggregation strategy to detect credit card fraud. *Expert system with application*, 39: 12650-12657.
[12] L. Breiman. Random forests. Machine Learning, (2001). Vol(45); 5–32.
[13] J. Piotr., A.M. Niall, J.D. Hand, C. Whitrow, J. David (2008). Off the peg and bespoke classifiers for fraud detection. *Computational Statistics and Data Analysis,* 52, pp:4521-4532.
[14] L. Qibei. & J. Chunhua. (2011). Research on Credit Card Fraud Detection Model Based on Class Weighted Support Vector Machine. *Journal of Convergence Information Technology*, 6(1), 62-68.
[15] S. Maes, K.Tuyls, B.Vanschoenwinkel, B.Manderick (1993). Credit card fraud detection using Bayesian and neural networks. *In Proceedings of the First International NAISO Congress on Neuro Fuzzy Technologies*, pages 261-270.
[16] E.W.T.Ngai, H.Yong., Y.H.Wong, Y.Chen, X. Sun (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50:559–569
[17] M. Zareapoor, Seeja, K.R. & M. Alam ,Afshar (2012).Analyzing Credit Card:Fraud Detection Techniques Based On Certain Design Criteria. *International Journal of Computer Application*, 52(3):35-42.
[18] http://www.damienfrancois.be/blog/files/modelperfcheatsheet.pdf

Journal of Big Data

RESEARCH Open Access

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

Emmanuel Ileberi[1*], Yanxia Sun[1] and Zenghui Wang[2]

*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. Oftentimes, 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.

$$y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n \tag{1}$$

$$q = \frac{1}{1 + e^{-y}} \tag{2}$$

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).

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

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

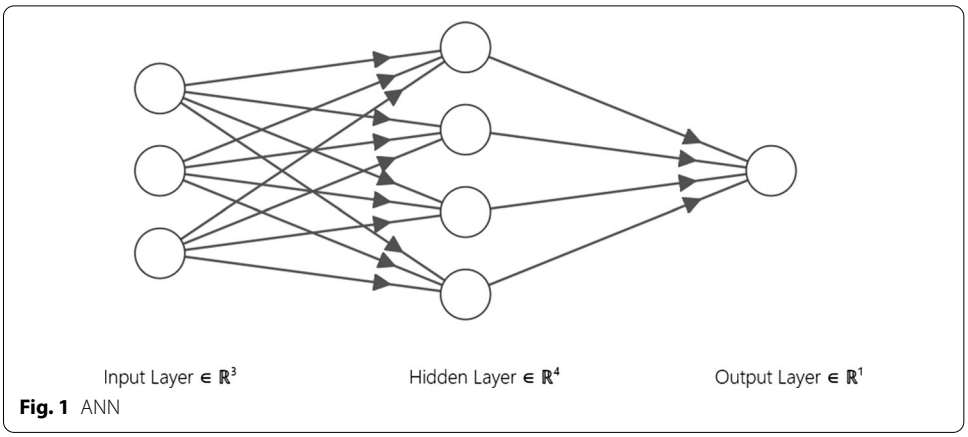


Fig. 1 ANN

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.

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.

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*,.., *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 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 **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 →(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].

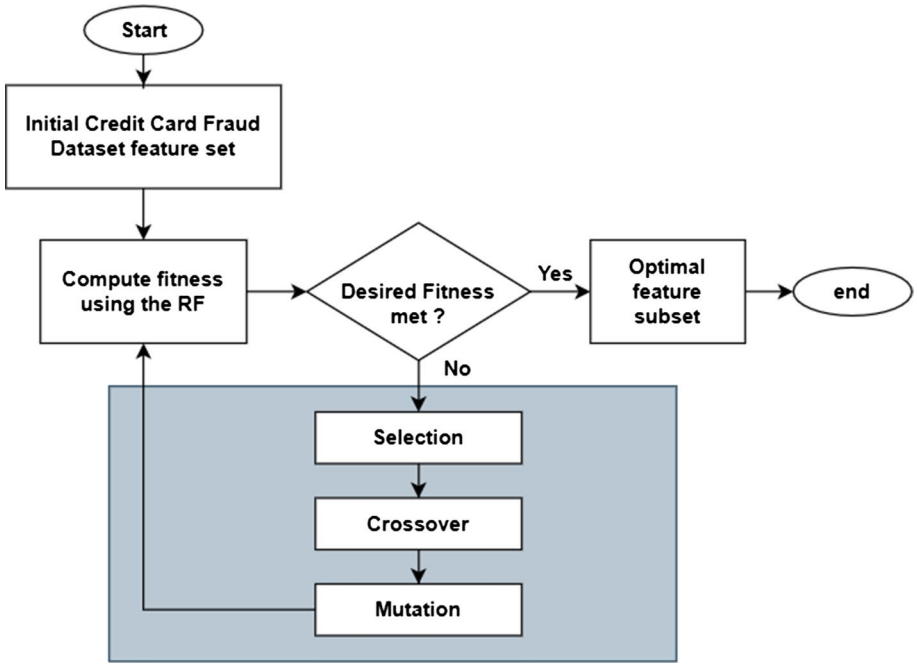


Fig. 2 GA flowchart

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

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

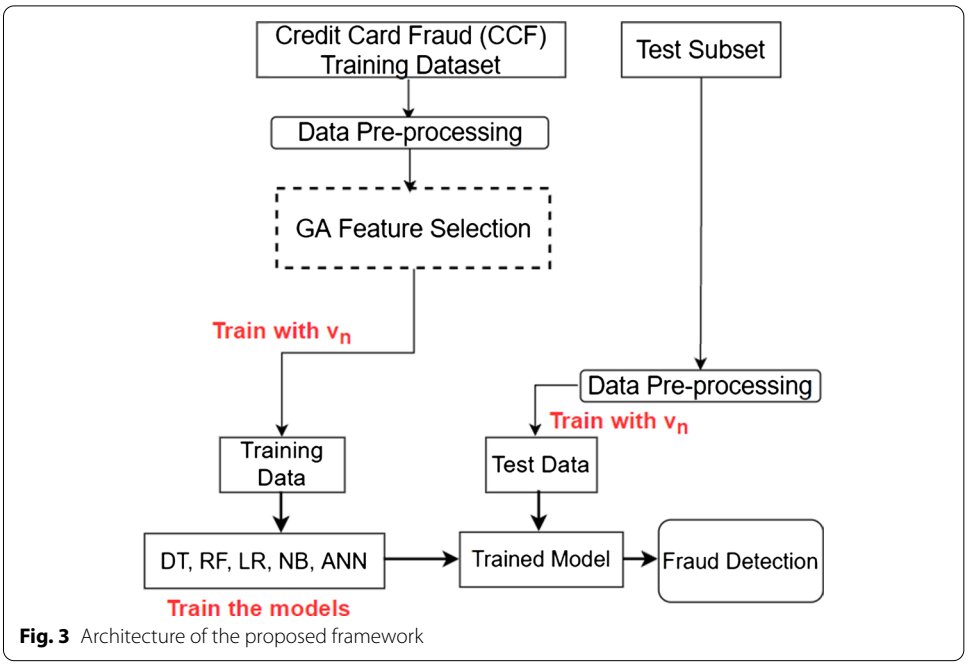


Fig. 3 Architecture of the proposed framework

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.

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

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 %

$$AC = \frac{TN + TP}{TP + TN + FP + FN} \tag{5}$$

$$RC = \frac{TP}{FN + TP} \tag{6}$$

$$PR = \frac{TP}{FP + TP} \tag{7}$$

$$F1_{score} = 2\frac{PR.RC}{PR + RC} \tag{8}$$

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 random_vec = { V2, V3, V4, V5, V6, V7, V8, V9, V11, V12, V13, V16, V17, V18, V19, V20, V21, V22, V23, V25, V26, V28, 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: $V = \{$ 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 %

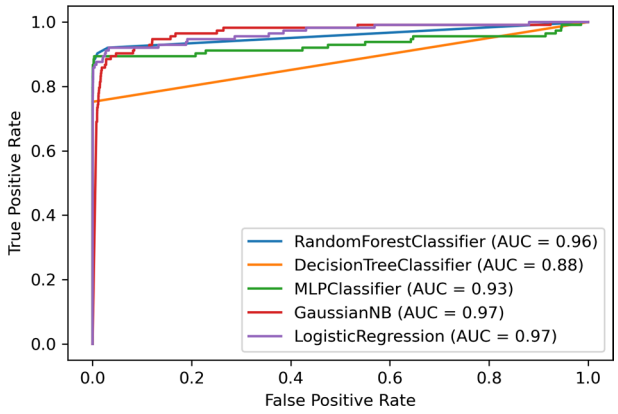


Fig. 4 AUC results for v_1

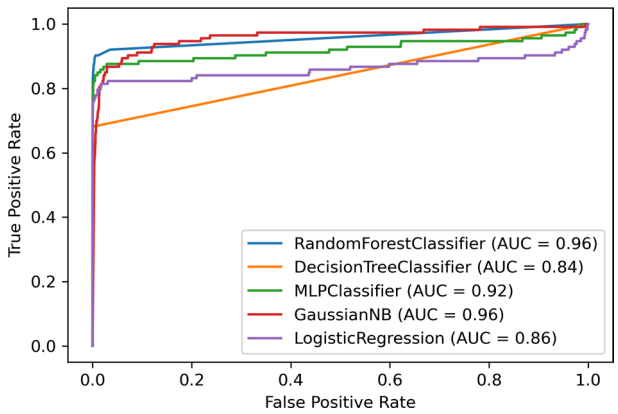


Fig. 5 AUC results for v_2

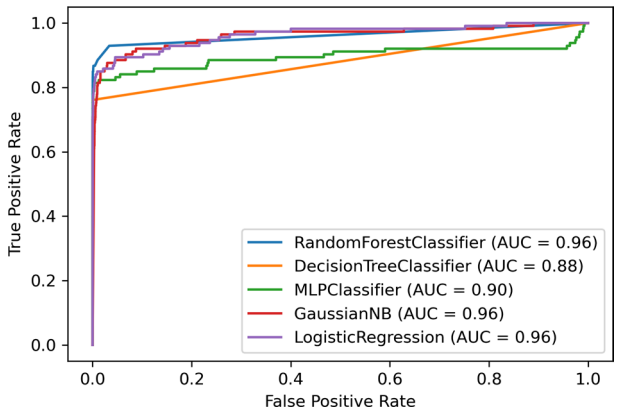


Fig. 6 AUC results for v_3

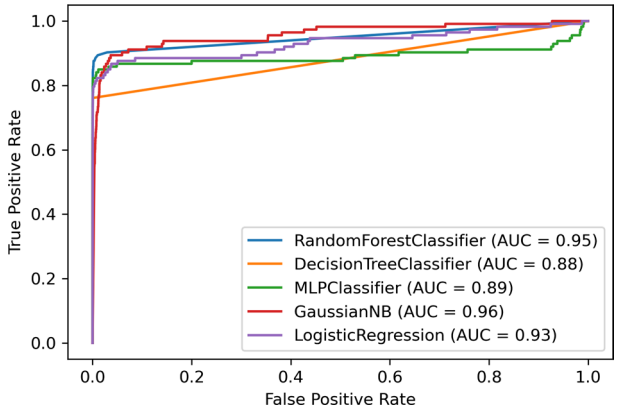


Fig. 7 AUC results for v_4

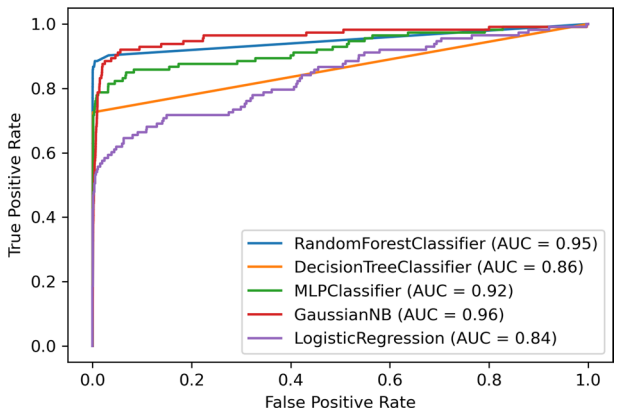


Fig. 8 AUC results for v_5

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). Here 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
[1]Department of Electrical & Electronic Engineering Science, University of Johannesburg, Kingsway Ave, 2006 Johannesburg, South Africa. [2]Department 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

1. 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.
2. Burkov A. The hundred-page machine learning book. 2019;1:3–5.
3. 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).
4. Dornadula VN, Geetha S. Credit card fraud detection using machine learning algorithms. Proc Comput Sci. 2019;165:631–41.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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 (ICCNI); 2017. p. 1-9.
17. Seera M, Lim CP, Kumar A, Dhamotharan L, Tan KH. An intelligent payment card fraud detection system. Ann Oper Res 2021;1–23.
18. 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.
19. The Credit card fraud [Online]. https://www.kaggle.com/mlg-ulb/creditcardfraud
20. Kasongo SM. An advanced intrusion detection system for IIoT based on GA and tree based algorithms. IEEE Access. 2021;9:113199–212.
21. Mienye ID, Sun Y. Improved heart disease prediction using particle swarm optimization based stacked sparse autoencoder. Electronics. 2021;10(19):2347.
22. 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.
23. Pouramirarsalani 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.
24. 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.
25. Davis L. Handbook of genetic algorithms; 1991.
26. 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.
27. 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.
28. 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.
29. 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

Publisher's Note

mathematics

MDPI

Article

Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest

Tzu-Hsuan Lin [1] and Jehn-Ruey Jiang [2,*]

[1] Department of Computer Science, University of Southern California, Los Angeles, CA 90007, USA; lintzuhs@usc.edu
[2] Department of Computer Science and Information Engineering, National Central University, Taoyuan City 320317, Taiwan
* Correspondence: jrjiang@csie.ncu.edu.tw

Abstract: This paper proposes a method, called autoencoder with probabilistic random forest (AE-PRF), for detecting credit card frauds. The proposed AE-PRF method first utilizes the autoencoder to extract features of low-dimensionality from credit card transaction data features of high-dimensionality. It then relies on the random forest, an ensemble learning mechanism using the bootstrap aggregating (bagging) concept, with probabilistic classification to classify data as fraudulent or normal. The credit card fraud detection (CCFD) dataset is applied to AE-PRF for performance evaluation and comparison. The CCFD dataset contains large numbers of credit card transactions of European cardholders; it is highly imbalanced since its normal transactions far outnumber fraudulent transactions. Data resampling schemes like the synthetic minority oversampling technique (SMOTE), adaptive synthetic (ADASYN), and Tomek link (T-Link) are applied to the CCFD dataset to balance the numbers of normal and fraudulent transactions for improving AE-PRF performance. Experimental results show that the performance of AE-PRF does not vary much whether resampling schemes are applied to the dataset or not. This indicates that AE-PRF is naturally suitable for dealing with imbalanced datasets. When compared with related methods, AE-PRF has relatively excellent performance in terms of accuracy, the true positive rate, the true negative rate, the Matthews correlation coefficient, and the area under the receiver operating characteristic curve.

Keywords: autoencoder; credit card; deep learning; fraud detection; data imbalance; random forest

Citation: Lin, T.-H.; Jiang, J.-R. Credit Card Fraud Detection with Autoencoder and Probabilistic Random Forest. *Mathematics* **2021**, *9*, 2683. https://doi.org/10.3390/math9212683

Academic Editors: Radu Tudor Ionescu and Guansong Pang

Received: 12 September 2021
Accepted: 21 October 2021
Published: 22 October 2021

1. Introduction

Credit card fraud is the unauthorized use of credit cards to obtain money, goods, or service by fraud. With the rise of e-commerce and contactless payment, credit cards are now widely used anywhere and anytime. For example, there are an estimated 1.1 trillion credit cards in the United States alone [1]. Thus, it is not surprising that millions of people fall victim to credit card fraud every year. Due to the prevalence of credit card transactions, cases of credit card fraud are also rampant globally. Americans reported 271,823 credit card fraud cases in 2019, an increase of 72.4% from 2018 [2]. Monitoring credit card transactions is not easy due to the large volume of data. Therefore, credit card fraud transactions are easily ignored, leading to huge losses for both cardholders and issuers. According to Nilson Report [3], credit card fraud caused losses of USD 28.65 billion in 2019, increasing by 2.9% from USD 27.85 billion in 2018. By 2020, global financial losses caused by credit card fraud amounted to USD 31 billion [4]. In addition to cardholders' vigilance and issuers' supervision, effective fraud detection methods must be adopted to detect credit card fraud automatically. This motivates the authors to develop credit card fraud detection methods based on advanced technologies.

Many credit card fraud detection methods [2,4–15] have been proposed in the literature. The readers are referred to two survey papers [16,17] for detailed descriptions of the methods. The survey paper [16] raised three challenging problems in credit card fraud

detection. The first is the data imbalance problem caused by the huge difference between the numbers of the positive and the negative classes. Specifically, as normal transactions far outnumber fraudulent transactions, credit card fraud detection methods are likely to overfit normal transactions. The second problem is the dataset shift, which means that fraud behaviors may evolve. New customer behaviors and new attacks on credit card transactions will deter fraud detection methods from maintaining good performance. The last problem is the oversight of sequential information among adjacent transactions. This is because investigators usually focus on a separate transaction but features of a separate transaction cannot reveal relations hidden among adjacent transactions. Different approaches are proposed to address different problems mentioned above. For instance, the papers [5,6] detailed the data shift problems and gave corresponding solutions to this problem. As for the ignorance of sequential information, it was tackled in the papers [7,8], which proposed methods having been tested to be effective. Some papers [9–12] utilized data resampling mechanisms to solve the data imbalance problem to have good fraud detection performance. However, some other papers [13–15] proposed methods that are naturally suitable for dealing with the data imbalance problem.

This research proposes a method, autoencoder with probabilistic random forest (AE-PRF), for credit card fraud detection. The proposed AE-PRF method first uses the autoencoder (AE) [18] to extract transaction data features. It then employs the random forest (RF) [19] with probabilistic classification to classify credit card transactions as normal or fraudulent. As just mentioned, the AE and the RF models can efficiently handle imbalanced data [20,21]. AE-PRF adopts the AE and the RF models since credit card transactions are typical imbalanced data. Moreover, unlike other methods adopting the RF with 0/1 classification, AE-PRF adopts the RF with probabilistic classification so that the performance of AE-PRF can be further improved, as will be shown later.

The credit card fraud detection (CCFD) dataset [22] released on the Kaggle platform was applied to AE-PRF for performance evaluation. The CCFD dataset contains credit card transactions of European cardholders within two days, including the normal transactions and the fraudulent transactions. It is extremely imbalanced, as the fraudulent data account for only 0.172% of total data. To make the CCFD dataset more balanced, data resampling schemes such as the synthetic minority oversampling technique (SMOTE) [23], adaptive synthetic (ADASYN) [24], and Tomek link (T-Link) [25] were applied to CCFD before data were fed into AE-PRF. As will be shown later, the performance of AE-PRF did not vary much whether resampling schemes were applied to the dataset or not. This indicates that AE-PRF is naturally suitable for dealing with imbalanced data. The performance of AE-PRF was compared with those of most related methods [12–15] that rely on CCFD for performance evaluation. Note that the methods proposed in [12] take or do not take data resampling, whereas the other methods proposed in [13–15] do not take data resampling. The performance comparisons are shown in terms of the accuracy, true positive rate, true negative rate, Matthews correlation coefficient, and area under the receiver operating curve to show the superiority of AE-PRF.

The contribution of the paper is threefold. First, it proposes AE-PRF that first uses AE to extract features of low-dimensionality from credit card transaction data features of high-dimensionality, and then relies on the RF with probabilistic classification to classify data as fraudulent or normal. By adopting the RF with probabilistic classification, the performance of AE-PRF can be improved. Second, experiments were conducted to apply data resampling schemes to imbalanced data before they were fed into AE-PRF. The experimental results show that AE-PRF is naturally suitable for dealing with imbalanced data, as the performance of AE-PRF does not vary much whether resampling schemes are applied to the data or not. Third, extensive experiments were conducted to evaluate the performance of AE-PRF and the performance evaluation results were compared with those of existing methods proposed in the literature [12–15]. The comparison results show that AE-PRF has relatively excellent performance in terms of accuracy, the true positive rate,

the true negative rate, the Matthews correlation coefficient, and the area under the receiver operating characteristic curve.

The rest of this paper is organized as follows. Section 2 describes the proposed AE-PRF method and some preliminaries. Section 3 then details related work. The performance evaluation of AE-PRF and its comparisons with related methods are shown in Section 4. Finally, Section 5 concludes this paper.

2. The Proposed Method

As just mentioned, the proposed AE-PRF method uses the AE to extract data features and employs the RF with probabilistic classification to classify credit card transactions as normal or fraudulent. To describe AE-PRF clearly, the concepts of the AE and the RF are first elaborated below.

2.1. Autoencoder

An AE [18] is a special type of artificial neural network that comprises connected neurons. Each neuron takes input vector x and generates the output y according to the following Equation (1):

$$y = \sigma\left(wx^{\mathrm{T}} + b\right), \tag{1}$$

where $\sigma(\cdot)$ is a nonlinear activation function (e.g., a sigmoid function), w is a weight vector, x^{T} is the transposition of x, and b is a bias vector.

The neural network structure of an AE is symmetric, as shown in Figure 1. An AE has one input layer, one or more hidden layers, and one output layer. Especially, the output layer of an AE has the same number of neurons as the input layer. Furthermore, the kth hidden layer and the $(n-k+1)$th hidden layer (or the kth hidden layer from the bottom) have the same number of neurons, where $k = 1, \ldots, \lfloor n/2 \rfloor$, and n is the number of hidden layers. The middle hidden layer is called the bottleneck, and the states (values) of neurons in the bottleneck layer constitute the code or the latent representation of the input. The code can be regarded as the extracted feature or the dimensionality reduction result of the original input.

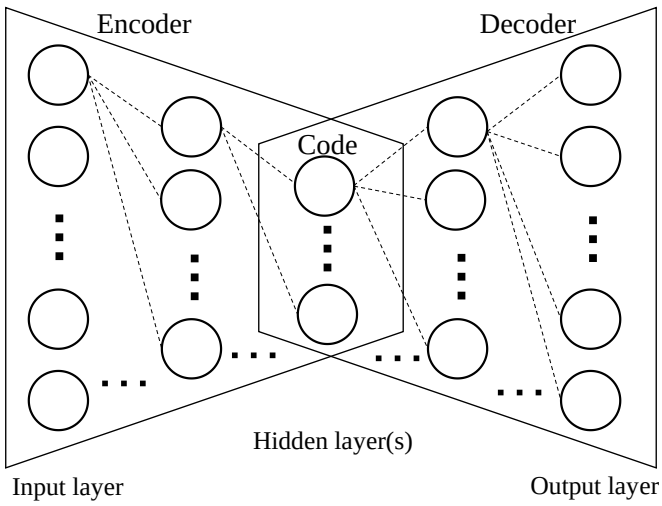


Figure 1. The neural network structure of an autoencoder (AE) model.

The first half part of an AE is called the encoder, whereas the second half part is called the decoder, as shown in Figure 1. The encoder encodes the input into the code, and the decoder decodes the code into the output. The output is intended to be as close to the input as possible; it is called the reconstructed input. The difference between the input and the output is called the reconstruction error. The AE is trained with the goal of minimizing the reconstruction errors by using the error backpropagation, gradient descent, and various optimizers like adaptive moment (Adam) optimizer.

2.2. Random Forest

The RF is an ensemble learning model for classification, regression, and other tasks [19]. Since the proposed AE-PRF method is for the task of classification, only the classification task is discussed in the following context. Specifically, the RF model utilizes decision trees to classify data and employs the bagging (i.e., bootstrap aggregating) approach to avoid the overfitting problem caused by complex decision trees. Below, the concept of using decision trees to classify data is first described.

A decision tree is a tree-like structure in which each internal node has a "split" based on an attribute, and each leaf node represents a prediction (or classification) result. Some metrics, such as the Gini impurity, entropy, and standard deviation, can be used for selecting the best splitting with the largest information gain. Below, the Gini impurity is taken as an example to show how the information gain is measured in decision trees. The information gain $IG(N_p, a)$ at node N_p split into c child nodes $N_1, \ldots, N_c$ based on the attribute a is defined in the following Equations (2) and (3):

$$IG(N_p, a) = Gini(N_p) - \sum_{i=1}^{c} \frac{|N_i|}{|N_p|} Gini(N_i) \tag{2}$$

$$Gini(N_p) = 1 - \sum_{j=1}^{m} p_j^2 \tag{3}$$

In Equation (2), $|N_p|$ stands for the number of data at node N_p, and $|N_i|$ stands for the number of data at node N_i, $0 \leq I \leq c$. In Equation (3), m is the number of different labels of data at node N_p, and p_j is the ratio of the number of data with the jth label over the total number of data at node N_p.

The best splitting with the largest information gain is performed for every possible attribute and every possible attribute value of dividing. The splitting continues until one of the following three stop conditions occurs. The three stop conditions are (i) all data at a node have the same label, (ii) the number of data at a node reaches a pre-specified minimum limitation, and (iii) the depth of a node reaches a pre-specified maximum limitation. After the splitting stops, the decision trees can be used to classify an input sample. The input sample goes through the tree from the root node to a leaf node, and it is classified as the label that dominates others at the leaf node.

Below we describe the bagging approach that randomly selects partial data and partial attributes to construct a variety of decision trees to be combined for data classification. This can avoid the overfitting problem that is intrinsic in decision trees. Given a dataset of d data or observations, the bagging approach produces n sub-datasets by drawing d' out of d observations with replacement, where $d' \leq d$. Every sub-dataset, along with a randomly selected subset of attributes, is used to train a decision tree. There are thus in total n decision trees that are trained independently with different sub-datasets and different attributes. Finally, either majority voting or averaging is applied to the n decision trees to get the final output of the RF, as shown in Figure 2.

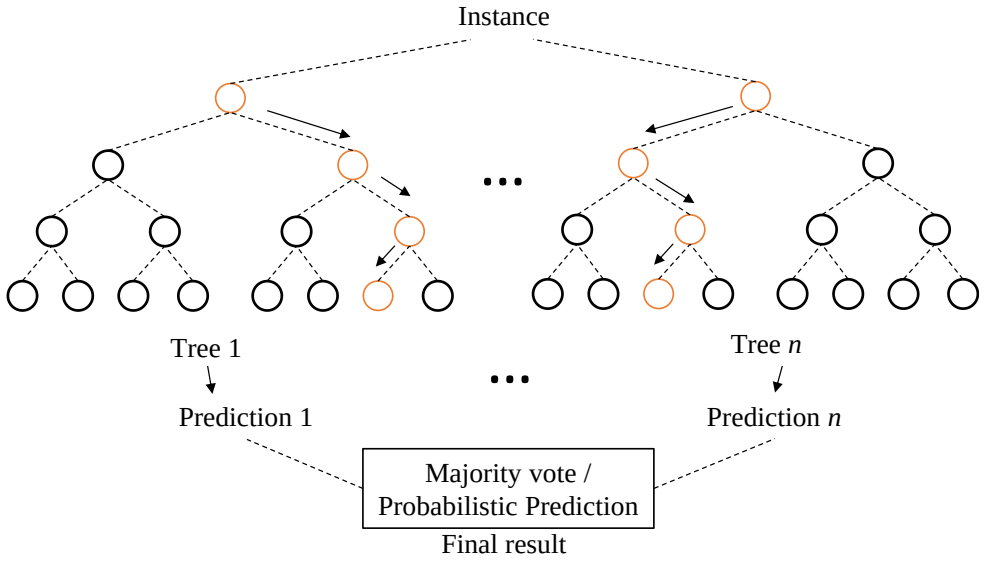


Figure 2. Illustration of a random forest (RF) model.

In general, an RF model can be used to classify an input instance into one of r classes $c_1, \ldots, c_r$. The procedure used to construct the RF model with n decision trees for a dataset of d data with k attributes has the following three major steps:

Step 1. Produce n sub-datasets from the original dataset of d data. Each sub-dataset is produced by drawing d' out of the d data with replacement, where $d' \leq d$.

Step 2. For each of the n sub-datasets, grow a decision tree by choosing the best splitting of internal tree nodes with the largest information gain for arbitrary k' attributes, $k' < k$. There are thus in total n decision trees to generate n classifications, each of which is one of r classes (i.e., labels) $c_1, \ldots, c_r$.

Step 3. Aggregate the results of the n trees to output the dominant class $c_{out} = \text{argmax}_{i=1}^{r} freq(c_i)$ as the final classification, where $freq(c_i)$ is the frequency that c_i appears among the n classifications. Note that the output may be adjusted to be with probabilistic classification, i.e., to output the classification frequencies (or probabilities) $freq(c_1), \ldots, freq(c_r)$ for all classes $c_1, \ldots, c_r$.

2.3. The Proposed AE-PRF Method

The proposed AE-PRF method first partitions the whole dataset as the training data, the validation data, and the test data. Figure 3 shows the processes of the proposed AE-PRF method. As shown in Figure 3, the data first undergo some preprocessing, and AE-PRF then applies the training data and the validation data to train an AE model. The AE model training is achieved by adjusting AE model weights properly with well-known error backpropagation and gradient descent mechanisms. The AE model can be used to reduce the data dimensionality and extract features from data as codes. Afterward, the codes of the training data are used to train an RF model to classify data into fraudulent data or normal data with associate classification probabilities. Moreover, the codes of the validation data are fed into the trained RF model to determine a proper threshold of classification probability to classify data with the best performance. Finally, for the verification purpose, the trained AE and RF models, along with the determined threshold can be applied to every test datum to check if it is fraudulent or normal.

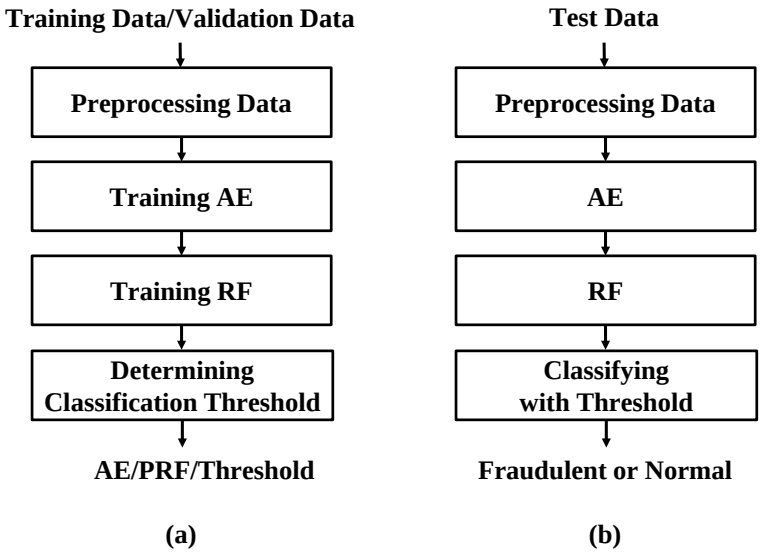


Figure 3. The illustration of AE-PRF: (**a**) the training process, and (**b**) the verification process.

Instead of directly using the RF model with a sole classification result, the AE-PRF method uses the RF model with probabilistic classifications to classify data. Specifically, the RF model with probabilistic classifications is used to classify a datum as fraudulent with probability p, and as normal with probability $1 - p$, where $0 \leq p \leq 1$. Afterward, AE-PRF outputs the final classification as fraudulent if p is larger than a pre-determined classification probability threshold θ. Different threshold values make AE-PRF generate different classification results. It is obvious that smaller θ values lead to a higher likelihood of classifying data as fraudulent. Fine-tuning the probability threshold θ value can provide AE-PRF with a customized classification result.

After the whole dataset is partitioned into the training data, the validation data, and the test data, the training process of AE-PRF can be started, as summarized in the following steps:

Step 1. Employ the training data to train the AE model AET and obtain the set T of training data feature codes.
Step 2. Train the RF model RFT with the set T of training data feature codes.
Step 3. Apply AET to the validation data to extract the set V of validation data feature codes.
Step 4. For threshold θ = 0 to 1 step s (=0.01), execute the following: Feed every code in V into RFT to output a probability p of fraud classification. If $p >$ θ, then the classification result is positive (fraudulent); otherwise, the classification result is negative (normal).
Step 5. Employ the classification results of all codes in V to find the threshold value θ* producing the best classification performance in terms of a specific metric M.

After the above-mentioned AE-PRF training process is finished, the verification process can be started, as summarized in the following steps.

Step 1. Apply AET to every test datum d to extract its feature code c.
Step 2. Feed the code c into RFT with the threshold value θ* to produce the classification result of d.

The pseudocode the proposed AE-PRF method is shown as Algorithm 1 below. The source code of AE-PRF implementation can be found at https://github.com/LinTzuHsuan/AE-PRF (accessed on 10 October 2021).

Algorithm 1 AE-PRF

Input: training data D_{train}, validation data $D_{validation}$, test data D_{test}, and metric M
Output: the classification result of each test datum (0 for normal or 1 for fraudulent)
Train the AE model AE_T with D_{train}
$T \leftarrow AE_T(D_{train})$
Train the RF model RF_T with T
$V \leftarrow AE_T(D_{validation})$
for $\theta \leftarrow 0$ to 1 step 0.01 **do**
    for each v in V **do**
        $p \leftarrow RF_T(v)$
        if $p > \theta$ **then** *result* $[\theta][v] \leftarrow 1$
        else *result* $[\theta][v] \leftarrow 0$
Find the best θ^* by comparing all *result* values in terms of metric M
$C \leftarrow AE_T(D_{test})$
for each c in C **do**
    $q \leftarrow RF_T(c)$
    if $q > \theta^*$ **then** *output* $[c] \leftarrow 1$
    else *output* $[c] \leftarrow 0$
return *output*

3. Related Work

The methods proposed in [12–15] are most related to AE-PRF. They all use the CCFD dataset for performance evaluation. None of them undergo data resampling except the methods proposed in [12]. Below, the related methods are elaborated one by one.

Three credit card fraud detection methods, namely naïve Bayes (NB), *k*-nearest neighbor (*k*-NN), and logistic regression (LR), are proposed in [12]. The best classification result is achieved by the *k*-NN method with $k = 3$. The *k*-NN method is a non-parametric supervised machine learning algorithm that can be used for classification and regression [26]. A test datum is classified into the dominant class of its *k* nearest neighbors' classes. Note that the random data resampling mechanism is adopted in [12] to address the data imbalance problem. Fraudulent data are oversampled and normal data are undersampled to make the ratio of fraudulent data to normal data 10:90 or 34:66 (≈1:2). The performance evaluation results show that data resampling can improve the performance of the *k*-NN method. However, it will be shown in this paper that data resampling does not necessarily improve the classification performance of the *k*-NN method.

Two unsupervised machine learning methods based on the AE model and the restricted Boltzmann machine (RBM) model are proposed in [13] for detecting credit card frauds. Like AE, RBM [27] can be sued to reconstruct input data. Both methods are unsupervised, as they need no data labels for training models. RBM can be regarded as a two-layer neural network with an input layer (visible) and a hidden layer. It is able to learn the probability distribution of the input data and thus can learn to reconstruct the data. This is achieved by fine-tuning the neural connection weights and biases through the processes of gradient descent and error back-propagation. For a new datum, either the trained AE or the trained RBM can be used to reconstruct the datum. The datum is assumed to be fraudulent if it has a large reconstruction error. As shown in [13], both AE and RBM have good fraud detection performance. However, AE is shown to have a better performance than RBM.

An unsupervised AE-based clustering method is proposed in [14] for detecting credit card frauds. The method uses an AE autoencoder with three hidden layers in both the encoder and the decoder. Moreover, it chooses the exponential linear unit (ELU) and the rectified linear unit (ReLU) as the activation functions of neurons in different layers. It also takes root mean square propagation (RMSProp) as the optimizer to yield the best result after performing several experiments. As shown in [14], the AE-base clustering method can achieve good classification performance by choosing an appropriate threshold of AE reconstruction errors to separate fraudulent data from normal data properly.

Twelve machine learning models for credit card fraud detection are studied in [15], including support vector machine (SVM) [28], naïve Bayes (NB), and feed-forward neural network (NN), etc. Furthermore, two ensemble learning mechanisms, namely adaptive boosting (AdaBoost) [29] and majority voting (MV), are combined with the twelve models to boost performance. Through comprehensive performance comparisons, SVM combined with AdaBoost (denoted as SVM + AdaBoost), and NN and NB combined with MV (denoted as NN + NB + MV) have comparably high performance. The SVM model generates a decision boundary in an increased or infinite-dimensional space, which is suitable for non-linear classification problems [30]. The AdaBoost method is an iterative method that adds a new weak classifier (i.e., classification model) in each iteration until all data are correctly classified, or the maximum iteration level has been reached. The NN + NB + MV model uses the feed-forward neural network and naïve Bayes concept [31] to perform fraud detection. The NN is an artificial neural network widely used in binary classification problems [32]. The NB is widely used for classification based on Bayes' theorem with strong independence assumptions between features [33]. It is good for the cases that the independence assumption fits. Due to MV, the NN + NB + MV model yields good classification results even when data are added with 10% to 30% of noise.

4. Performance Evaluation and Comparisons

4.1. Dataset and Data Resampling

The CCFD dataset [22] contains data generated by European cardholders within 2 days in September 2013. It has a total of 284,807 transactions, among which 492 are fraudulent. The dataset is highly imbalanced because fraudulent data account for 0.172% of total data. Each data entry has 31 attributes, including the transaction timestamp, the transaction amount, and the transaction class or label, which is 1 if the transaction is fraudulent, and 0, otherwise. It also has 28 principal component analysis (PCA) transformation values of transaction data. The PCA values are transformed from transaction data. They are for the purpose of hiding information like the cardholder identity and personal privacy data. Note that PCA is a feature extraction mechanism to project high-dimensional data into low-dimensional data without losing crucial information. It can also be used to transform data for the purpose of data dimensionality reduction, data feature extraction, and data de-identification.

As mentioned earlier, in order to make the CCFD dataset more balanced, data resampling schemes such as SMOTE [23], ADASYN [24], and T-Link [25] are applied to CCFD data before they are fed into AE-PRF. The three schemes are used to balance the numbers of majority class samples (or majority samples, for short) and minority class samples (or minority samples, for short). Their basic ideas are described below.

SMOTE is an oversampling technique. For a minority sample x_i, SMOTE first finds k nearest minority samples based on the k-NN scheme. It then selects a sample x_j out of the k nearest minority samples and generates a new minority sample x_{new} according to the equation: $x_{new} = x_i + \delta(x_j - x_i)$, where $\delta \epsilon [0, 1]$. The process to generate new minority samples continues until the number of newly generated minority samples reaches the pre-specified value.

ADASYN is also an oversampling technique. It is similar to SMOTE, but it adaptively generates new minority samples for a minority sample according to its imbalance degree. Specifically, for a minority sample x_i, its k nearest samples are first derived and its imbalance degree is defined as Δ_i/k, where Δ_i is the number of majority samples out of the k nearest samples of x_i.

T-Link is an undersampling technique. It tries to find a pair of a minority sample x_i and a majority sample x_j such that there is no sample x_k satisfying $d(x_k, x_j) < d(x_i, x_j)$ or $d(x_i, x_k) < d(x_i, x_j)$, where $d(u, v)$ is the Euclidean distance between samples u and v. It then removes the majority sample of every such pair so that the boundary between the majority class and the minority class is clearer and hence samples are easier to be classified.

4.2. Performance Metrics

The performance evaluation metrics, accuracy (ACC), the true positive rate (TPR), the true negative rate (TNR), and the false positive rate (FPR) are defined below in Equations (4)–(7), respectively.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{4}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{6}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{7}$$

In Equations (4)–(7), TP, FP, TN, and FN stand for the numbers of true positive, false positive, true negative, and false negative classifications (or predictions), respectively. A positive prediction is the one classifying a transaction as fraudulent, whereas a negative prediction is the one classifying a transaction as normal (i.e., not fraudulent). TP (respectively, FP) is the number of positive predictions for fraudulent (respectively, normal) transactions. TN (respectively, FN) is the number of negative predictions for normal (respectively, fraudulent) transactions. Note that TPR is also called sensitivity or recall, TNR is also called specificity, and FPR is also called the false alarm rate.

The area under the receiver operating characteristic curve (AUC) is a metric related to the receiver operating characteristic (ROC) curve. The ROC curve can be used as a tool to consider the tradeoff between TPR and FPR for a classifier based on threshold values. Different threshold values lead to different TPRs and FPRs. The ROC curve can be plotted by setting the x-axis as FPR and the y-axis as TPR, and the area under the ROC curve is then AUC. Larger AUC values correspond to better classifiers. If AUC has a value of 0.5, then the classifier is a no-skill classifier. If AUC has a value of 1, then the classifier is perfect.

The Matthews correlation coefficient (MCC) [34], as defined in Equation (8), can be regarded as a comprehensive metric, since it addresses TP, FP, TN, and FN at the same time. MCC has values within the range between -1 and $+1$, where the value of $+1$ indicates perfect predictions and -1 means entirely conflicting predictions. As stated in [35], MCC is suitable for both balanced and imbalanced datasets. Therefore, in the following performance evaluation, we consider MCC as an important metric.

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{8}$$

Several evaluation metrics, including ACC, TPR, TNR, MCC, AUC, recall, F1-score, log loss/binary cross-entropy, and categorical cross-entropy, can be used for evaluating the performance of classification methods. Among them, ACC, TPR, TNR, and MCC, which are defined in Equations (4)–(6) and (8), as well as AUC are commonly used for evaluating the performance of credit card fraud detection methods [12–15]. Specifically, the two most important metrics are TPR and MCC. The reason for the first metric, TPR, to be adopted in fraud detection is that the higher the TPR, the more fraudulent data can be detected, which is the main purpose of fraud detection. However, when evaluating the averaging performance of a model, MCC would be considered, because it takes every parameter including TP, TN, FP, and FN into consideration. To sum up, ACC, TPR, TNR, MCC, and AUC are deployed as metrics for performance evaluations and comparisons.

4.3. Performance Evaluation of AF-PRF

To evaluate the performance of the proposed AE-PRF method, the CCFD dataset is first partitioned into a training dataset of 64% data, a validation dataset of 16% data, and a test dataset of 20% data. All data undergo pre-processing such as the logarithmic transform on the amount of transaction and the second-to-day transform on the number

of seconds elapsed between the transaction and the first transaction in the dataset. The hyper-parameters of AE-PRF are described as follows. The AE model has five hidden layers with 26, 20, 18, 20, and 26 neurons, respectively, using the rectified linear unit (ReLU) as the activation function. The AE model uses the Adam as the optimizer, except for the first layer, using hyperbolic tangent (Tanh) instead. In order to prevent overfitting, L1 Regularization is applied to the first layer of the encoder. At the training stage, early stopping is adopted to prevent overfitting, using validation loss as the monitor. The dimension of the CCFD dataset data is reduced from 26 attributes to 18 by the trained AE. Well-defined feature extraction and dimensionality reduction algorithms (e.g., the AE model) make the detection/classification process more effective and efficient. The most important and influential features of the data will be focused on after dimensionality reduction.

The RF model of AE-PRF has 100 decision trees (estimators) and uses Gini impurity as the criterion, as defined in Equation (3). It generates probabilistic classification, i.e., it classifies the test datum as fraudulent with probability p, $0 \leq p \leq 1$.

The AE-PRF performance evaluation has two parts. The first part does not apply resampling mechanisms to data, whereas the second part applies resampling mechanisms to data. Below, we first describe the first part.

As mentioned earlier, AE-PRF uses the RF model with probabilistic classification with probability p to check if a test datum is classified as fraudulent. If p is larger than a pre-specified classification threshold θ, then the test datum is assumed to be fraudulent. Certainly, different threshold values lead to different classification performances. In order to find the best threshold confronting different requirements, it is necessary to fine-tune and shift the threshold and find the one which produces the best result in terms of specific metrics. More specifically, fine-tuning the threshold by testing different threshold values 0, 0.01, 0.02, . . . , 1 in agreement with the evaluation metric is the way to find the best threshold.

The threshold is first obtained by the ROC curve. To be precise, 101 different threshold θ values are applied to the AE-PRF classifier, ranging from 0 to 1 with the step interval of 0.01. Experiments are conducted 50 times to derive the average TPR and FPR, which in turn are used to plot the ROC curve, as shown in Figure 4. The zoomed-in version of Figure 4 is also given in Figure 5. We randomly repartitioned the dataset into a training set, a validation set, and a test set, and repeat the experiment 50 times to reduce biases of experimental results. The diagonal line in Figure 4 indicates the curve for a no-skill classifier. The upper left point on the ROC curve in Figure 4 indicates a model with perfect skill, which is computed by the geometric mean (or g-mean) of TPR and FPR (i.e., $\sqrt{\text{TPR} \times (1 - \text{FPR})}$. The g-mean of TPR and FPR is a good indicator of classification for imbalanced data. When it is optimized, a balance between the sensitivity (i.e., TPR) and specificity (i.e., TNR) is reached. The threshold recommended by the ROC curve is 0.03, which is the one corresponding to the best g-mean. The AUC of the ROC curve is 0.962, which is better than 0.960 of the method proposed in [13] and 0.961 of the method proposed in [14].

Similarly, the threshold is then tuned to obtain the best ACC, TPR, TNR, and MCC, as demonstrated in Figure 6. The ACC is very high with 101 thresholds, all about 0.99, except for $\theta = 0$. However, a high ACC alone cannot be interpreted as this fraud detection classifier being good enough. When dealing with highly imbalanced datasets, it is common to get a high ACC [36]. Therefore, other evaluation metrics must be taken into consideration. As for TNR, it also gets high scores with most of the thresholds, and the highest score is around 0.9998 achieved by $\theta = 0.25$. However, for the same reason as the ACC metric, because datasets of fraud detection problems are usually highly imbalanced, it tends to obtain a much higher TN value than FP value, which easily results in a high TNR [37].

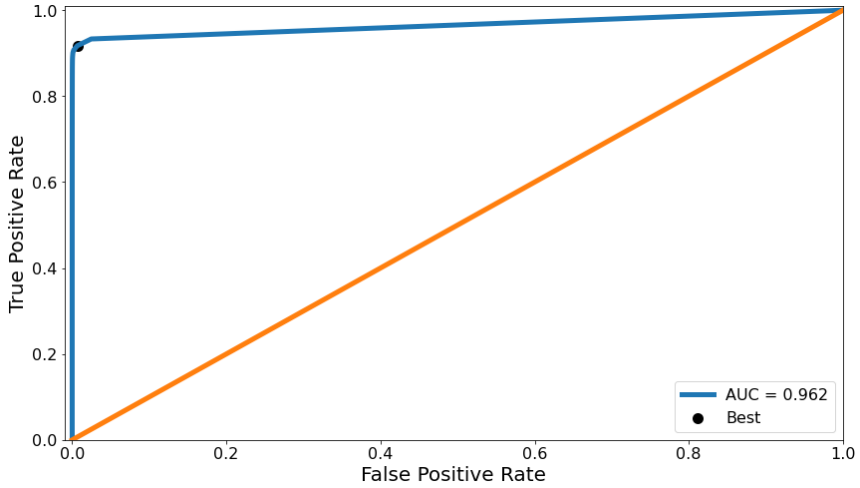


Figure 4. The ROC curve of AE-PRF.

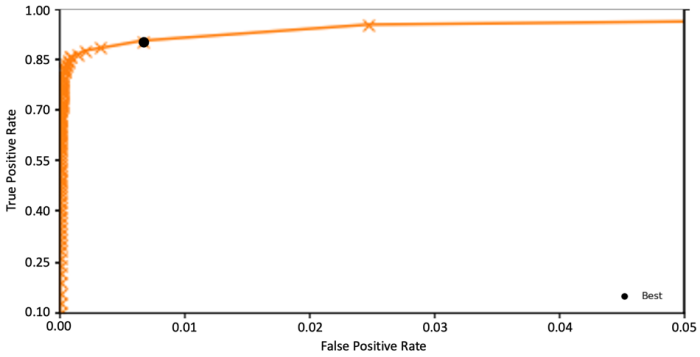


Figure 5. The zoomed-in ROC curve of AE-PRF (for FPR in [0.00, 0.05] and TPR in [0.01, 1.00]).

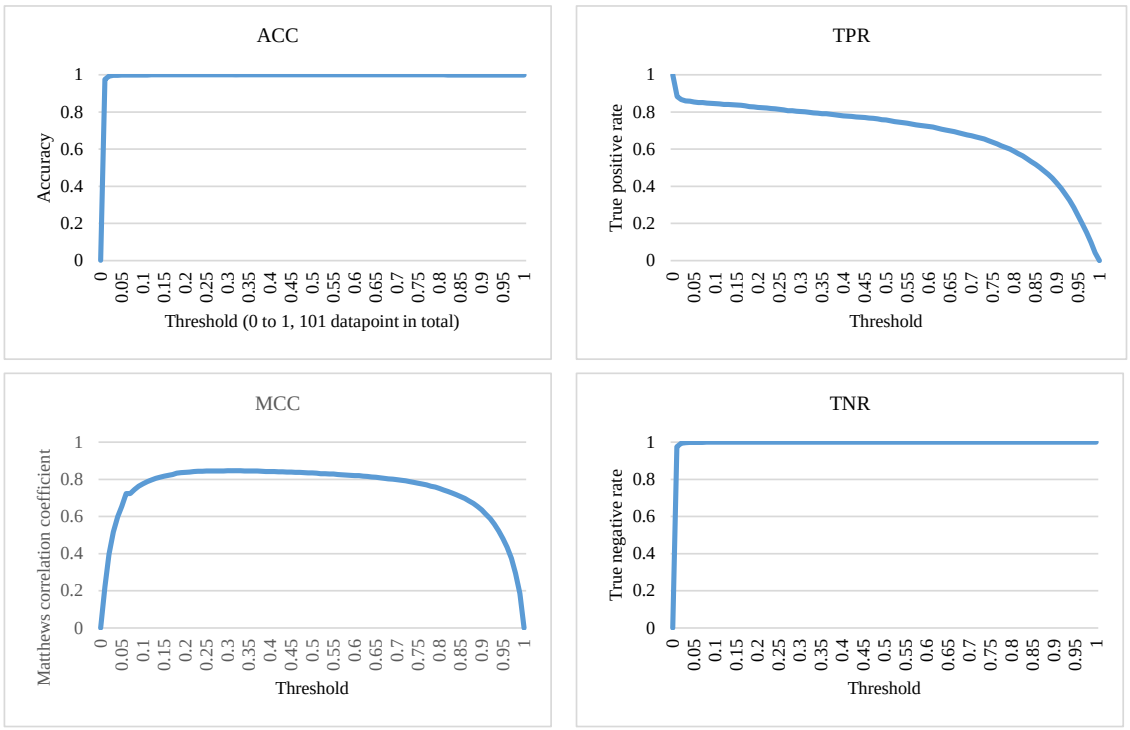


Figure 6. The ACC, TPR, TNR, and MCC of AE-PRF for different threshold values.

Therefore, this study considers MCC as a very important metric. Specifically, when considering and choosing the optimal threshold, this study just focuses on the thresholds generating MCC scores that are greater than 0.8. Thus, only threshold values ranging from 0.13 to 0.69 are considered. Consequently, the best average MCC is around 0.8456, obtained by setting $\theta = 0.25$.

However, it is risky to consider only one metric. In the credit card fraud detection problem, it is desirable to achieve higher TPRs so that more fraudulent data can be detected [36]. Nonetheless, if there are no restrictions, the highest TPR will always be achieved by $\theta = 0$. Under such a setting, not only ACC and TNR but also MCC will be very low, being 0. Thus, here, the premise is to set MCC greater than 0.5 and find the best TPR. In other words, if we want to detect as many fraudulent transactions as possible while maintaining a decent overall performance, another threshold must be adopted. To the best of our knowledge, the TPR of 0.89109 achieved by setting θ as 0.03 is the highest one ever seen while keeping MCC greater than 0.5. Note that setting θ as 0.03 is also recommended by the ROC curve using the g-mean of TPR and FPR.

Table 1 shows the details of performance metrics of AE-PRF for $\theta = 0.25$, which yields the best average MCC in our experiments. The details are the average score, the lowest scores (Minimum), the first quartile (Q1), the second quartile (Q2), the third quartile (Q3), and the highest scores (Maximum) of ACC, TPR, TNR, and MCC in 50 experiments with $\theta = 0.25$. The average score will be compared with those of other related methods later.

Table 1. Performance of AE-PRF in 50 times of experiments ($\theta = 0.25$).

Metrics	Average	Minimum	Q1	Q2	Q3	Maximum
ACC	0.99949	0.999410129	0.999455774	0.999494396	0.999522485	0.999578664
TPR	0.8142	0.75	0.808333333	0.816666667	0.825	0.84166667
TNR	0.9998	0.999704567	0.999788976	0.999803044	0.999831181	0.999887454
MCC	0.8441	0.811201026	0.834295395	0.845629405	0.853475139	0.870180134

Now, the second part of the AE-PRF performance evaluation is described. In this part, ADASYN itself and the combination of SMOTE and T-Link, denoted as SMOTE + T-Link, were applied to the training dataset for resampling data to make them more balanced. This was to verify whether the use of data resampling techniques can improve AE-PRF performance. However, if data resampling does not noticeably improve AE-PRF performance, then AE-PRF is said to be naturally suitable for dealing with imbalanced data.

The performance evaluation results of the AE-PRF without data resampling and with ADASYN and SMOTE + T-Link data resampling are shown in Table 2. The sampling strategies of ADASYN and SMOTE + T-Link are the same. That is, the ratio of the minority sample quantity over the majority sample quantity is set to 34:66 ($\approx$1:2) for both ADASYN and SMOTE + T-Link. Specifically, both ADASYN and SMOTE + T-Link adjust the number of fraudulent transactions to be 142,172, and the number of normal transactions to be 284,315 for the training dataset. As observed from Table 2, the performance of AE-PRF is not noticeably improved by data resampling. Moreover, AE-PRF using no data resampling even has better performance than AE-PRF using data resampling in terms of some metrics. It thus may be proper to say that AE-PRF is naturally suitable for dealing with imbalanced data.

Table 2. Performance of AE-PRF with and without data resampling.

Models	ACC	TPR	TNR	MCC
AE-PRF (θ = 0.03)	0.9973	0.8910	0.9975	0.5921
AE-PRF (θ = 0.25)	0.9995	0.8142	0.9998	0.8441
ADASYN AE-PRF (θ = 0.13)	0.9960	0.8613	0.9963	0.5018
ADASYN AE-PRF (θ = 0.57)	0.9995	0.8316	0.9998	0.8665
SMOTE + T-Link AE-PRF (θ = 0.11)	0.9965	0.8583	0.9967	0.5133
SMOTE + T-Link AE-PRF (θ = 0.51)	0.9995	0.8333	0.9998	0.8585

4.4. Perforamance Comparisons

Here, after several experiments, θ is set to be of values 0.25 and 0.03 for comparing AE-PRF and five related methods in terms of various performance metrics to demonstrate the superiority of AE-PRF. The five methods are the *k*-NN [12], AE [13], AE based clustering [14], SVM + AdaBoost [15], and NN + NB with MV [15]. All methods for comparison, including the proposed AE-PRF, use no data sampling. The performance comparisons were performed in terms of ACC, TPR, TNR, AUC, and MCC.

Table 3 shows the performance comparison results of AE-PRF and other five related methods. The highest scores in Table 3 are in boldface. It can be seen that AE-PRF outperformed others in almost all metrics. As for AE-PRF with θ = 0.25, it had the highest ACC of 0.9995, the highest TNR of 0.9998, and the highest MCC of 0.8441. However, its TPR of 0.8142 was lower than the highest score of 0.8835 achieved by *k*-NN [12]. Therefore, another threshold was adopted, AE-PRF with θ = 0.03 had the highest TPR of 0.89109 and comparable high ACC, TNR, and MCC. If the main goal of the credit card fraud detection is to achieve as high TPR as possible while maintaining a decent MCC (say $\geq$ 0.5), then AE-PRF with θ = 0.03 is the best one to choose.

Table 3. Performance comparisons of AE-PRF and related methods.

Research	Methods	ACC	TPR	TNR	MCC	AUC
Awoyemi et al. [12]	*k*-NN	0.9691	0.8835	0.9711	0.5903	-
Pumsirirat et al. [13]	AE	0.97054	0.83673	0.97077	0.1942	0.9603
Zamini et al. [14]	AE-based clustering	0.98902	0.81632	0.98932	0.3058	0.961
Randhawa et al. [15]	SVM with AdaBoost	0.99927	0.82317	0.99957	0.796	-
Randhawa et al. [15]	NN+NB with MV	0.99941	0.78862	0.99978	0.823	-
This Research	AE + PRF (θ = 0.03)	0.99738	**0.89109**	0.99757	0.5921	**0.962**
This Research	AE + PRF (θ = 0.25)	**0.9995**	0.8142	**0.9998**	**0.8441**	**0.962**

Note that the *k*-NN method proposed in [12] has two versions, one using data resampling and the other using no data resampling. However, only the version using no data resampling is compared with the proposed AE-PRF method in Table 3. This is because when we re-implement the *k*-NN method and apply random data resampling to the re-implemented *k*-NN method, the performance of the re-implemented *k*-NN does not conform with the performance results shown in [12]. As demonstrated in Table 4, the data resampling even makes *k*-NN have bad performance. The research [12] likely applied data resampling to the whole data, including the training and the test data, whereas we apply data resampling to only the training data. We confirm this by applying random data resampling to the whole data and then running the re-implemented *k*-NN method. As observed from Table 4, if the whole data is resampled, then the performance results of the original *k*-NN and the re-implemented *k*-NN are quite similar. However, not all

test data can be obtained in advance and each test datum should be classified separately. Resampling all data, including training data and test data, seems to be impractical.

Table 4. Performance of *k*-NN [12] and re-implemented *k*-NN with and without data resampling.

Methods	ACC	TPR	TNR	MCC
k-NN [12] (without resampling)	0.9691	0.8835	0.9711	0.5903
k-NN [12] (with all data 34:66 resampling)	0.9792	0.9375	1.0	0.9535
Re-implemented *k*-NN (without resampling)	0.9977	0.7483	0.9981	0.5512
Re-implemented *k*-NN (with only training data resampling)	0.9817	0.1881	0.9832	0.0556
Re-implemented *k*-NN (with all data 34:66 resampling)	0.9832	0.9494	1.0	0.9624

5. Conclusions

This paper proposes a fraud detection method called AE-PRF. It employs AE to reduce data dimensionality and extract data features. Moreover, it utilizes RF with probabilistic classification to classify data as fraudulent along with an associated probability. AE-PRF outputs the final classification as fraudulent if the associated probability exceeds a predetermined probability threshold θ.

The CCFD dataset [22] was applied to evaluate the performance of AE-PRF. Since the CCFD dataset is highly imbalanced, data resampling schemes like SMOTE [23], ADASYN [24], and T-Link [25] were applied to the CCFD dataset to balance the numbers of normal and fraudulent transactions. Experimental results showed that the performance of AE-PRF does not vary much whether resampling schemes are applied to the dataset or not. This indicates that AE-PRF is naturally suitable for handling imbalanced datasets without data resampling.

The performance evaluation results of AE-PRF without data resampling were compared with those of related methods such as *k*-NN [12], AE [13], AE-based clustering [14], SVM with AdaBoost [15], and NN + NB with MV [15]. The comparison results show that AE-PRF with $\theta = 0.25$ has the highest ACC, TNR, MCC, and AUC, and has comparably high TPR. As for AE-PRF with $\theta = 0.03$, it has the highest TPR and AUC, and comparable high ACC, TNR, and MCC. The CCFD dataset is partitioned into a training dataset of 64% data, a validation dataset of 16% data, and a test dataset of 20% data for evaluating AE-PRF performance. We tried another extreme partition, a training dataset of 40% data, a validation dataset of 10% data, and a test dataset of 50%, which does not yield a good result because of the insufficient training data.

It is more persuasive to compare AE-PRF to existing methods using the same dataset for performance evaluation. Since the CCFD dataset was adopted by many existing methods and it is the most detailed public dataset, this paper adopted the CCFD dataset for performance evaluation and comparison. However, in order to test the robustness and effectiveness of AE-PRF, we need to adopt some other datasets, especially private datasets, because there are few public datasets for credit card fraud detection due to privacy issues. In the future, we plan to cooperate with credit card issuers and/or banks to obtain datasets for verifying the robustness and the effectiveness of AE-PRF.

In the future, we will try to improve AE-PRF performance by fine-tuning the hyperparameters of the AE and the RF models. We will also try to apply AE-PRF to a variety of applications for evaluating AE-PRF's applicability. Furthermore, we will investigate the explainability of AE-PRF and try to enhance AE-PRF's explainability by leveraging novel explainable AI (XAI) schemes proposed in [38–40] for AE and RF.

Author Contributions: Conceptualization, T.-H.L. and J.-R.J.; funding acquisition, J.-R.J.; investigation, T.-H.L. and J.-R.J.; methodology, T.-H.L. and J.-R.J.; software, T.-H.L.; supervision, J.-R.J.; validation, T.-H.L. and J.-R.J.; writing—original draft, T.-H.L. and J.-R.J.; writing—review & editing, T.-H.L. and J.-R.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology (MOST), Taiwan, under the grant number 109-2622-E-008-028-.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. de Best, R. Credit Card and Debit Card Number in the U.S. 2012–2018. Statista. 2020. Available online: https://www.statista.com/statistics/245385/number-of-credit-cards-by-credit-card-type-in-the-united-states/#statisticContainer (accessed on 10 October 2021).
2. Voican, O. Credit Card Fraud Detection using Deep Learning Techniques. *Inform. Econ.* **2021**, *25*, 70–85. [CrossRef]
3. The Nilson Report. Available online: https://nilsonreport.com/mention/1313/1link/ (accessed on 20 December 2020).
4. Taha, A.A.; Sharaf, J.M. An intelligent approach to credit card fraud detection using an opti-mized light gradient boosting machine. *IEEE Access* **2020**, *8*, 25579–25587. [CrossRef]
5. Dal Pozzolo, A. Adaptive Machine Learning for Credit Card Fraud Detection. Ph.D. Thesis, Université Libre de Bruxelles, Brussels, Belgium, 2015.
6. Lucas, Y.; Portier, P.-E.; Laporte, L.; Calabretto, S.; Caelen, O.; He-Guelton, L.; Granitzer, M. Multiple perspectives HMM-based feature engineering for credit card fraud detection. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, ACM, New York, NY, USA, 8–12 April 2019; pp. 1359–1361.
7. Wiese, B.; Omlin, C. Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks. In *Studies in Computational Intelligence*; Springer Science and Business Media LLC: Berlin, Germany, 2009; pp. 231–268.
8. Jurgovsky, J.; Granitzer, M.; Ziegler, K.; Calabretto, S.; Portier, P.-E.; He-Guelton, L.; Caelen, O. Sequence classification for credit-card fraud detection. *Expert Syst. Appl.* **2018**, *100*, 234–245. [CrossRef]
9. Zhang, F.; Liu, G.; Li, Z.; Yan, C.; Jiang, C. GMM-based Undersampling and Its Application for Credit Card Fraud Detection. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
10. Ahammad, J.; Hossain, N.; Alam, M.S. Credit Card Fraud Detection using Data Pre-processing on Imbalanced Data—Both Oversampling and Undersampling. In Proceedings of the International Conference on Computing Advancements, New York, NY, USA, 10–12 January 2020; ACM Press: New York, NY, USA, 2020.
11. Lee, Y.-J.; Yeh, Y.-R.; Wang, Y.-C.F. Anomaly Detection via Online Oversampling Principal Component Analysis. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 1460–1470. [CrossRef]
12. Awoyemi, J.O.; Adetunmbi, A.O.; Oluwadare, S.A. Credit card fraud detection using machine learning techniques: A comparative analysis. In Proceedings of the 2017 International Conference on Computing Networking and Informatics (ICCNI), Lagos, Nigeria, 29–31 October 2017; pp. 1–9.
13. Pumsirirat, A.; Yan, L. Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 18–25. [CrossRef]
14. Zamini, M.; Montazer, G. Credit Card Fraud Detection using autoencoder based clustering. In Proceedings of the 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 17–19 December 2018; pp. 486–491. [CrossRef]
15. Randhawa, K.; Loo, C.K.; Seera, M.; Lim, C.P.; Nandi, A.K. Credit Card Fraud Detection Using AdaBoost and Majority Voting. *IEEE Access* **2018**, *6*, 14277–14284. [CrossRef]
16. Lucas, Y.; Johannes, J. Credit card fraud detection using machine learning: A survey. *arXiv* **2020**, arXiv:2010.06479.
17. Nikita, S.; Pratikesh, M.; Rohit, S.M.; Rahul, S.; Chaman Kumar, K.M.; Shailendra, A. Credit card fraud detection techniques—A survey. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering, Vellore, India, 24–25 February 2020.
18. Rumelhart, D.E.; Geoffrey, E.H.; Ronald, J.W. Learning internal representations by error propagation. *Calif. Univ. San Diego La Jolla Inst. Cogn. Sci.* **1985**, *8*, 318–362.
19. Liaw, A.; Matthew, W. Classification and regression by random Forest. *R News* **2002**, *2*, 18–22.
20. Seeja, K.R.; Zareapoor, M. FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining. *Sci. World J.* **2014**, *2014*, 1–10. [CrossRef]
21. Zhang, C.; Gao, W.; Song, J.; Jiang, J. An imbalanced data classification algorithm of improved autoencoder neural network. In Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 95–99.
22. Credit Card Fraud Detection Dataset. Available online: https://www.kaggle.com/mlg-ulb/creditcardfraud (accessed on 20 August 2020).
23. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
24. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the IEEE International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 1322–1328. [CrossRef]
25. Tomek, I. Two Modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 769–772. [CrossRef]
26. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185.

27. Sutskever, I.; Geoffrey, E.H.; Graham, W.T. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*; University of Toronto: Toronto, ON, Canada, 2009.
28. Bhattacharyya, S.; Jha, S.; Tharakunnel, K.; Westland, J.C. Data mining for credit card fraud: A comparative study. *Decis. Support Syst.* **2011**, *50*, 602–613. [CrossRef]
29. Margineantu, D.; Dietterich, T. Pruning Adaptive Boosting. In Proceedings of the 14th International Conference on Machine Learning, ICML, Guangzhou, China, 18–21 February 1997.
30. Naveen, P.; Diwan, B. Relative Analysis of ML Algorithm QDA, LR and SVM for Credit Card Fraud Detection Dataset. In Proceedings of the 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 7–9 October 2020; pp. 976–981.
31. Rish, I. An Empirical Study of the Naive Bayes Classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. 2001, Volume 3. No. 22. Available online: https://www.cc.gatech.edu/fac/Charles.Isbell/classes/reading/papers/Rish.pdf (accessed on 20 August 2020).
32. Jeatrakul, P.; Wong, K.W. Comparing the performance of different neural networks for binary classification prob-lems. In Proceedings of the 2009 Eighth International Symposium on Natural Language Processing, Bangkok, Thailand, 20–22 October 2009.
33. Murphy, K.P. *Naive Bayes Classifiers*; University of British Columbia: Vancouver, BC, Canada, 2006.
34. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 1–13. [CrossRef] [PubMed]
35. Saito, T.; Rehmsmeier, M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE* **2015**, *10*, e0118432. [CrossRef] [PubMed]
36. Chawla, N.V. Data Mining for Imbalanced Datasets: An Overview. *Data Min. Knowl. Discov. Handb.* **2009**, 875–886. [CrossRef]
37. Lakshmi, T.J.; Prasad, C.S.R. A study on classifying imbalanced datasets. In Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), Guntur, India, 19–20 August 2014; pp. 141–145.
38. Assaf, R.; Giurgiu, I.; Pfefferle, J.; Monney, S.; Pozidis, H.; Schumann, A. An Anomaly Detection and Explainability Framework using Convolutional Autoencoders for Data Stor-age Systems. *IJCAI* **2020**, 5228–5230. [CrossRef]
39. Antwarg, L.; Miller, R.M.; Shapira, B.; Rokach, L. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. *Expert Syst. Appl.* **2021**, *186*, 115736. [CrossRef]
40. Fernández, R.R.; de Diego, I.M.; Aceña, V.; Fernández-Isabel, A.; Moguerza, J.M. Random forest explainability using counterfactual sets. *Inf. Fusion* **2020**, *63*, 196–207. [CrossRef]