I.J. Information Technology and Computer Science, **2021, 4, 24-37**
Published Online August 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2021.04.03

Credit Card Fraud Detection System Using Machine Learning

Angela Makolo and Tayo Adeboye
Department of Computer Science, University of Ibadan, Nigeria
E-mail: aumakolo@gmail.com, tayo.nifes@gmail.com

Received: 27 February 2021; Revised: 03 April 2021; Accepted: 19 April 2021; Published: 08 August 2021

Abstract: The security of any system is a key factor toward its acceptability by the general public. We propose an intuitive approach to fraud detection in financial institutions using machine learning by designing a Hybrid Credit Card Fraud Detection (HCCFD) system which uses the technique of anomaly detection by applying genetic algorithm and multivariate normal distribution to identify fraudulent transactions on credit cards. An imbalance dataset of credit card transactions was used to the HCCFD and a target variable which indicates whether a transaction is deceitful or otherwise. Using F-score as performance metrics, the model was tested and it gave a prediction accuracy of 93.5%, as against artificial neural network, decision tree and support vector machine, which scored 84.2%, 80.0% and 68.5% respectively, when trained on the same data set. The results obtained showed a significant improvement as compared with the other widely used algorithms.

Index Terms: Credit card fraud, multivariate Gaussian distribution, genetic algorithm, artificial neural network, decision tree, support vector machine.

1. Introduction

Credit card unarguably plays an important part in business, household and global activities. Although Credit card as a form of electronic banking helps to provide better customer services**,** despite the huge benefits of its deployment and usage, a significant amount of damage is caused by fraudulent activities **[**1,2**].** Financial services institutions are most at risk in the negative impact of fraud, despite the various protection mechanisms in place [3]. In a report issued by the Federal Trade Commission of the United States, 21 percent increase was recorded in cases of identity theft in 2008, though it had remained fairly steady during the mid-2000s. In 1999 alone, one out of every 1200 transactions of the 12 annual billion transactions made, approximating to about 10 million, turned out to be fraudulent.

In another report issued in 2014 by Lexis Nexis, the United States alone accounts for about half of the $11 billion yearly cases of fraudulent credit card transactions worldwide. A persistent increase in credit card fraud in Nigeria over a period of 3 years, from 2014-2016 was established by [4]. In the research, internet banking, ATM and web channels consummated top three fraudulent transactions in 2014, while POS, ATM, and web channels were top three in 2015, and in 2016, ATM, mobile and web channels topped the list. It can be inferred that ATM related fraud is not only consistent but also ranks highest among the most perpetrated fraudulent activities.

Fraud prevention and fraud detection are two major effective strategies that have been used in tackling fraud [5]. Measures taken to mitigate fraud is referred to as 'Fraud Prevention' while the ability to identify or discover fraudulent activities is referred to as 'Fraud Detection'. The difference between the two is that fraud detection takes place after the occurrence of a fraudulent activity whereas the other is aimed at preventing the occurrence. Fraud detection is a subject pertinent to several industries ranging from banking sectors, insurance, law enforcement and other government and private agencies [6]. In recent years, case of fraud has drastically increased, making fraud detection even more crucial than ever [7].

The continuous advancement in Internet technology and computing power has their positive and negative impact in the financial institution; while the positive impact includes that customers can now easily carry out wild-range of financial transaction at their convenience and the negative impact is that lost or stolen credit card or credit card information can easily be used by unauthorized person for fraudulent operations. Furthermore, as computing power keep growing so will the rate of credit card fraud since people will rely more and more on computerized process for their daily activities. Hence, there are needs for more accurate and reliable approach for detecting credit card fraud which will help to condense this illegal activity to the lowest minimum. Hence, this research work is focused on developing an efficient and reliable credit card fraud detection system.

A machine learning classifier model that can detect fraudulent and non-fraudulent credit card transaction using anomaly detection and genetic algorithm is proposed. We also evaluate the model and compared its performance with the performance of other existing algorithms (support vector machine, decision tree and neural network). At the end, the

best algorithm will be adopted using certain performance metrics in the work.

The continuing part of this work is arranged thus. Section II describes the related works, the classification algorithm, training and testing dataset in section III. Our experiments, result and analysis are discussed in Section IV while conclusions are drawn in section V.

2. Related Works

Efforts have been made to detect anomalies in credit card transactions using various machine learning techniques. Early works identified types of credit card fraud and various alternative techniques employed for fraud detection. In their work they outlined common terms and key statistics and figure in the field of credit card fraud. The types of fraud highlighted in their work are theft fraud/counterfeit fraud, bankruptcy fraud, behavioral fraud and application fraud. According to their research pair-wise matching, decision tree, genetic algorithm [8,9], clustering and neural network were the various techniques used for identifying credit card fraud.

[9] analyzed the working principles and the performance of a total of nine (9) machine learning algorithm which are Neural network, Bayesian Network, Support Vector Machine, K-Nearest Neighbor algorithm, Fuzzy logic based system, Decision tree, Hidden Markov Model, Artificial Immune System, and Genetic Algorithm (GA). When viewed for the given metrics, Bayesian Network seemed to have the highest score

Investigation on how issues of credit card fraud can be addressed using machine learning algorithms was done by [10]. The study focused on a framework that can report the transactions with the highest risk, employing algorithms that can deal with unbalanced and evolving data streams.

An inherent problem with the normal hidden Markov model was identified by [11] and hence, they proposed an Advanced Hidden Markov Model (AHMM), which proved to be more efficient in detecting credit card fraud than the already existing model.

[12] Proposed a detection model which uses aggregated transactions to learn consumer's buying behaviour before each transaction and then use the aggregations for model estimation to detect fraudulent transaction. This model was evaluated using sensitivity and specificity measure which showed high accuracy.

[13] Proposed a model using K-clustering and Hidden Markov Model (HMM) to detect credit card fraud. In this model, card holder's profile is categorized by HMM as low, medium and high spending based on their spending behavior, with amount put into consideration. A set of probabilities for amount of transaction was assigned to cardholders and number of incoming transactions was then matched with each card owner's category, if it justified a predefined threshold value, the transaction was decided to be legitimate, else it was declared fraudulent.

3. Methodology

3.1. Multivariate Normal Distribution

Statistical-based techniques uses parametric methods for detecting outliers and assume some knowledge of the distribution (normal or Gaussian in our case) of the dataset. For a normal distribution the farther a point deviate from the mean the less probable if becomes and hence, data points whose probabilities are lower than the threshold value in the model is considered as outliers. In this research a Multivariate Gaussian Distribution is the technique used in detecting anomaly behavior in credit card transaction since an observation is not a univariate variable.

For a univariate normal distribution, the probability density function is given as:

$$P(X;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)\right] \quad (1)$$

Equation 1 is used when X is a univariate random variable. Where X is a multivariate variable or observation, we say

$$\boldsymbol{X}_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1d} \end{bmatrix} \boldsymbol{X}_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2d} \end{bmatrix} \ldots\ldots\ldots \boldsymbol{X}_N = \begin{bmatrix} x_{N1} \\ x_{N2} \\ \vdots \\ x_{Nd} \end{bmatrix} \quad (2)$$

Then the mean $\boldsymbol{\mu}$ of the observations will be a vector given as:

$$\mu_i = \frac{1}{N}\sum_{j}^{d} x_{ji} \quad (3)$$

$$\text{So the } \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \cdot \\ \vdots \\ \mu_d \end{bmatrix} \tag{4}$$

Therefore the covariance of the observation is given as:

$$cov(X_j, X_k) = \frac{1}{N}\sum_{i}^{d}(x_{ji} - \mu_j)(x_{ki} - \mu_k) \tag{5}$$

Then the covariance $\sum$ matrix is

$$\sum = \begin{pmatrix} cov(X_1, X_1) & \cdots & cov(X_1, X_d) \\ \vdots & \ddots & \vdots \\ cov(X_d, X_1) & \cdots & cov(X_d, X_d) \end{pmatrix} \tag{6}$$

Hence the general normal or multivariate distribution is given as

$$P(X; \mu, \sum) = \frac{1}{(2\pi)^{\frac{d}{2}}|\sum|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(X-\mu)^T \sum^{-1}(X-\mu)\right] \tag{7}$$

3.2. Genetic Algorithm

Genetic Algorithm was first introduced in 1973 by John Holland in his publication on genetic algorithms and optimal allocation of trials. The concept is based on the survival of the fittest strategy during sexual reproduction as proposed by Charles Darwin [14]. When dealing with finding optimal solutions and stochastic search [15], then genetic algorithm is the best bet. These algorithms relies on the combination of chromosome populations, selection, crossover, and mutation [16] to produce new offspring. [17] summarizes the steps followed by GA to arrive at optimal solutions as follows:

1. Random generation of Initial population.
2. Fitness function for each chromosome is calculated, based on the fitness function predefined.
3. Two parents with the highest fitness are selected for crossover or mutation operators.
4. A new chromosome is added to the next generation.
5. Step 3 is reiterated until the size of the previous generation is equal to the next generation
6. Step 2 is reiterated until the stop condition is applied"

Some of the methods used for selecting the best chromosomes in the iteration, as seen in [18,19] are elitism and stochastic universal selection, rank selection, roulette wheel selection, steady state selection, tournament selection, truncation selection, and so on. The selection process ensures that most fit parents are selected and used in the new offspring, discarding the unfit parents, while the crossover involves taking a cross point randomly to swap the substrings from the parents to form two new offspring. Mutation operator enhances the GA to find optimal solutions and it works by changing some of the bits in the offspring to form the chromosome in the new population. The iteration is continued until all offspring have been created. [19].

3.3. Methodology Framework

This research method is divided into six (6) stages as shown in the generic framework, Fig. 1.

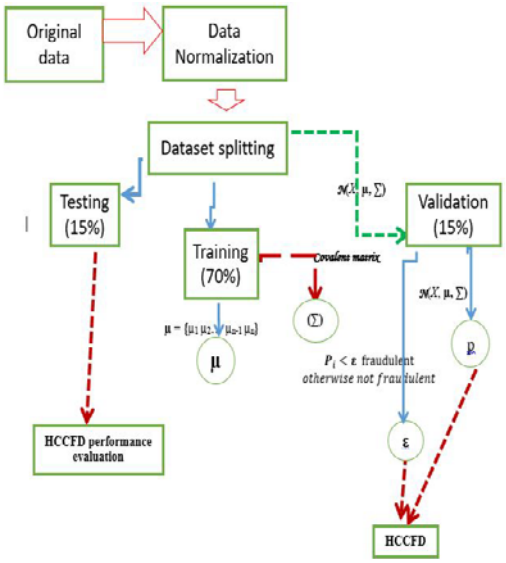

Fig.1. Generic framework for HCCFD

Step 1: The whole dataset which contains thirty thousand (30,000) observation was randomly divided into seventy percent (70%), fifteen percent (15%) and fifteen percent (15%) as training dataset, validating dataset and testing dataset respectively.

Step 2: The training dataset was used to compute the mean vector (μ) which is a column vector with each entry corresponding to the mean a column in the training dataset. That is

$$\mu = \{\mu_1\ \mu_2 \ldots\ \mu_{n-1}\ \mu_n\} \tag{8}$$

Where n is the number of features

Also the covariant matrix (∑) of training dataset was computed, which is an n by n matrix. That is

$$\sum = \begin{pmatrix} \sum_{1\,1} & \cdots & \sum_{1\,n} \\ \vdots & \ddots & \vdots \\ \sum_{n\,1} & \cdots & \sum_{n\,n} \end{pmatrix} \tag{9}$$

μ and ∑ were computed using MATLAB.

Step 3: Normal multivariate distribution N(X, μ, ∑) were used to compute probability vector (P) of all the observation (X) in the validation dataset. That is,

$$\mathrm{P} = \{\mathrm{P}_1\ \mathrm{P}_2 \ldots\ \mathrm{P}_{i-1}\ \mathrm{P}_i\} \text{ and } Pi\ =\ N(Xi, \mu, \sum) \tag{10}$$

Where i = the number of observation in the validation dataset.

Step 4: Genetic Algorithm (GA) was used to select ε (a real number) that minimizes the misclassification rate of the validation dataset such that

$$\begin{cases} \boldsymbol{P_i < \varepsilon} \text{ fraudulent} \\ otherwise\ not\ fraudulent \end{cases} \tag{11}$$

Step 5: N(X, μ, ∑) and ε are bundled to form a machine learning classifier model (called Hybrid Credit Card Fraud Detection HCCFD model) such that for an observation x

P= N(x, μ, ∑) then

$$\begin{cases} \boldsymbol{P < \varepsilon} \text{ fraudulent} \\ otherwise\ not\ fraudulent \end{cases} \tag{12}$$

Step 6: Support vector machine, artificial neural network and decision tree were also trained using the same training dataset.

The performance of the model was evaluated using the testing dataset and a comparison of the result was analyzed.

By making use of the multivariate distribution function and GA to search and select the threshold below which an observation is termed outlier (fraudulent), errors due to misclassification is thus minimized.

A. Data Collection

A dataset of thirty thousand (30,000) observations of credit card transactions was collected from a financial institution (name withheld, for privacy issues). This dataset contains features about the credit card and card holder. Of the features, nineteen features as shown in figure 3 were selected after much review and consultation from professionals in the banking sector. A target variable which could be zero (0) if the transaction is not fraudulent or one (1) if it is fraudulent was also adopted in the research.

B. Data Normalization

In data science, normalization is used during the preprocessing stage of data. It is used to find new range of data from an existing one [20], by reducing variations in the data, thus making them appear closer to each other, in a well-behaved range. There are several normalization techniques such as Z-score normalization, Min-Max normalization and Decimal scaling normalization [20]. Min-Max and Z-score are the most commonly used.

- **Min-Max Normalization**: In this technique, given the range ***[L,U]*** (typically between 0 and 1), each feature with value *x* is normalized in terms of the minimum and maximum values, $\boldsymbol{x_{max}}$ and $\boldsymbol{x_{min}}$, respectively, by fitting them in a predefined boundary, using the formula:

$$\boldsymbol{x'} = \frac{(\boldsymbol{x}-\boldsymbol{x_{min}})}{(\boldsymbol{x_{max}}-\boldsymbol{x_{min}})} \times (\boldsymbol{U}-\boldsymbol{L}) + \boldsymbol{L} \tag{13}$$

- **Z-Score Normalization**: Z-score also gives values ranging between 0 and 1. 0 for mean and 1 for standard deviation. This technique is also called standardization. The feature values are auto-transformed to get the mean and standard deviation. It follows this procedure for transformation: for each feature *f*, the mean value μ(*f*) and standard deviation σ(*f*) are calculated and then the feature with value *x* is transformed using the formula:

$$\boldsymbol{x'} = \frac{(\boldsymbol{x}-\mu(f))}{\sigma(f)} \tag{14}$$

The dataset obtained has nineteen (19) features and it was used to develop the proposed model, however Sixteen out of this nineteen (19) features were numeric and remaining three are categorical. The seventeen numerical features were normalized using equation 9 so as to get an optimal model.

$$x_i' = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{15}$$

Where x_i' is the new value for the ith entry of X feature (column) with $\max(X)$ and $\min(X)$ as maximum and minimum value, respectively.

In order to achieve a model with a high accuracy, the dataset was normalized using the following command in MATLAB

```
newX=normc(X);
```
(16)

where X= dataset.

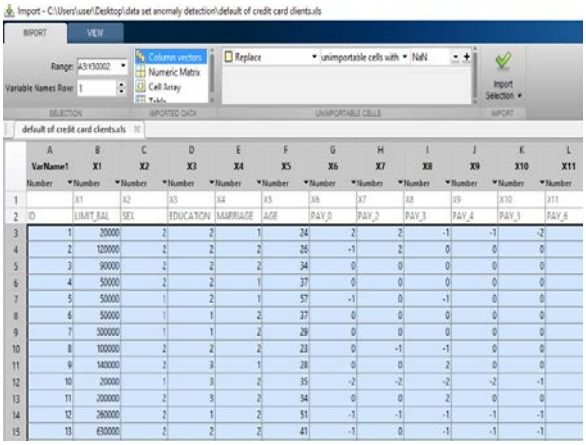

Fig.2. Dataset After Normalization

C. Input Data Transformation

This research, after consulting with professionals in the financial institution having reviewed the literature by [21], removed features that does not enhance the prediction model and hence, adopted the following 19 features as variables:

X1: Sex (*1 = male; 2 = female*).
X2: Marital status (*1 = married; 2 = single; 3 = others*).
X3: Educational level Categorical (Postgraduate = 1, university = 2, secondary school = 3, and others = 4).
X4: Age (Numeric).
X5: Credit limit (Numeric).
X6-X11: History of monthly spending for the last six months.
X12-X17: History of monthly payment for the last six months.
X18: Actual Amount spent
X19: the time the transaction was carried out.

D. Dataset splitting

To avoid contaminating the dataset of 30,000 observations was divided in the ratio 70%:15%:15% as training, testing and validation dataset respectively.

The training dataset is used to compute the Mean vector and covariate matrix while the validation dataset is used by GA to find the best threshold for an outlier and final, the test dataset is used to estimate the accuracy of the model.

3.4. Model Training phases

In developing the proposed model two major phases were followed and these are:

- Model parameter computing phase

To understand the pattern in the given dataset (training dataset), two important statistical parameter are needed, which are:

1. **Mean vector (μ):** This in computed by the use of equation 11 below.

$$\mu_i = \frac{1}{N}\sum_{j}^{d} x_{ji} \tag{17}$$

Where:

μ_i = the mean of the *ith* column (feature) in the training dataset. x_{ji} = the entry in the *jth* row of *ith* column in the training dataset.

N = the number of observation (row) in the training dataset.

j=1, 2, 3,….d, d is the number of row in training dataset. i=1,2,3…number of feature in the training dataset.

Finding pattern in a multivariate dataset involve computing mean vector of the dataset. This is done using the training dataset in MATLAB as follows:

meu= mean(X_train_set);

2. **Covariate matrix** (∑)**:** This is also computed by the use of equation 18 and 19 below.

$$cov(\boldsymbol{X}_j, \boldsymbol{X}_k) = \frac{1}{N}\sum_i^d (x_{ji} - \mu_j)(x_{ki} - \mu_k) \tag{18}$$

$$\sum = \begin{pmatrix} cov(X_1, X_1) & \cdots & cov(X_1, X_d) \\ \vdots & \ddots & \vdots \\ cov(X_d, X_1) & \cdots & cov(X_d, X_d) \end{pmatrix} \tag{19}$$

Where:

$\boldsymbol{X}_j$ = the *jth* observation in training dataset
$\boldsymbol{X}_k$ = the *kth* observation in training dataset
$cov(\boldsymbol{X}_j, \boldsymbol{X}_k)$ = the covariance of $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$.
x_{ji} = the entry in the *jth* column of *ith* row in the training dataset.
x_{ki} = the entry in the *kth* column of *ith* row in the training dataset.
μ_j = the mean of the *jth* column (feature) in the training dataset.
μ_k = the mean of the *kth* column (feature) in the training dataset.
i=1, 2, 3,….*d*, *d* is the number of row in training dataset.
j=1,2,3…number of feature in the training dataset.
k=1,2,3…number of feature in the training set.
N = the number of observation (row) in the training dataset.
$\sum$ = Covariate matrix.

A. Threshold selection phase

The threshold is a numerical value which serves as separating boundary between fraudulent and non-fraudulent transaction. Therefore, an optimal selection of this value is crucial for achieving an optimal dictation model (HCCFD).

If a particular transaction maximum likelihood estimate is less than ε the transaction will be classified as anomaly (fraudulent), hence, it will be classified as a legitimate transaction. In order to select the most appropriate threshold, Genetic Algorithm (GA) was employed which often guarantees optimal solution. The following steps was taken to determine the value of the threshold **ε**:

1. Computing the maximum likelihood estimate vector (P) if all the observation in the validation dataset as shown in equation 14

$$p_i = \frac{1}{(2\pi)^{\frac{d}{2}}|\sum|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(x_i - \mu)^T \sum^{-1}(x_i - \mu)\right] \tag{20}$$

Where:

x_i = the i[th] observation in the validation dataset.

2. GA is used to select value of **ε**, computed using the objective function f(**ε**).

```
f(ε):
   P← maximum_likelihood_estimate_vector
   Y← actual_target
   Y'←[ ]
   For i←1:length(p)
   If( Y[i]< ε):
   Y'.append(1)
   Else:
   Y'.append(0)
   C←confusion_matrix(Y,Y')
   Precision←C.precision()
   Recall←C.recall()
   Fscore←(2* Precision* Recall)/( Precision + Recall)
   Return (- Fscore)
```

The threshold is selected such that it maximizes classification accuracy of the actual target of validation dataset and its predicted probability using **N** (X_validation_set,meu,Z). This process is an iteration and a searching process which can best be done using GA.

3.5. Performance metrics derived from confusion matrix:

Accuracy: How often is the classifier correct, generally?

$$\frac{TP + TN}{total}$$

False Positive Rate: How often does it predict yes when it is actually no?

$$\frac{FP}{TN + FP}$$

Misclassification Rate: How often is it wrong, generally?

$$\frac{FP + FN}{total}$$

Precision: How often is it correct when it predicts yes?

$$\frac{TP}{FP + TP}$$

Specificity: How often does it predict no when it is actually no?

$$\frac{TN}{TN + FP}$$

True Positive Rate or Sensitivity or Recall: How often does it predict yes when it is actually yes?

$$\frac{TP}{FN + TP}$$

F Score: This is a weighted average of the true positive rate (recall) and precision.

$$\frac{2 \times Precision \times Recall}{Precision + Recall}$$

Figure 3 depicts the flow chart of the propose model at training stage in which the dataset is first split and mean vector and covariate matrix computed using training dataset and GA is used to select the optimal threshold for classification.

4. Results and Analysis

4.1. Analysis

A. Confusion Matrix

In machine learning, the summary of prediction results on a classification problem is represented using confusion matrix. The matrix shows the ways in which the classification model is confused during the prediction making process. It gives a general insight on the way the model makes errors and the types of errors being made

B. Decision Tree

In supervised machine learning, decision trees are used to explain what input and the corresponding output is in a classification. Here, the data is continuously fragmented according to a certain parameter. The decision tree mainly explains the decision nodes (where the data is split from) and the leaves (results or final outcome)

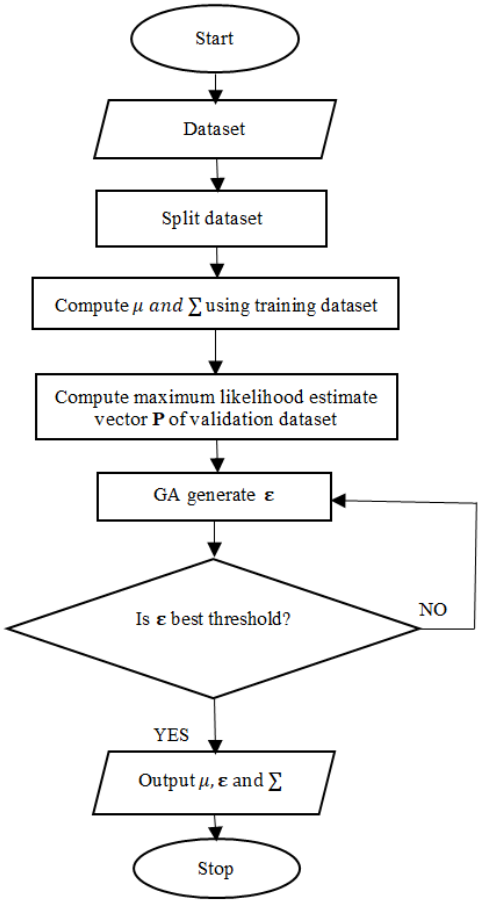

Fig.3. Flow chart of the proposed model at training stage

a. Receiver Operating Characteristics

A Receiver operating characteristics is a curve that shows the diagnostic ability of binary classifiers. It shows the trade-off between sensitivity and specificity. Classifiers that give curves closer to the top-left corner in receiver operating characteristics, indicate a better performance.

4.2. Training Support Vector Machine (SVM)

Support Vector Machines (SVM) has in the past proven to be very successful in classification problems [22]. In training the SVM, we consider two kernel (linear and Radian Bayesian Function - RBF). Table. 1 shows the result obtained using linear and RBF kernel. The confusion matrix of the various SVM model performances on the test dataset were plotted as shown in Fig. 4 and 5.

Table 1. Performance comparison of linear kernel SVM and RBF kernel SVM

	Linear kernel SVM	**RBF kernel SVM**
Number of support vectors	20458	19716
Bias	1.2213	0.1678
Accuracy	58.2%	58.2%
Misclassification rate	41.8 %	41.8 %
Recall	54.2%	54.2%
Specificity	79%	79%
Precision	93.1%	93.1%
F score	68.5%	68.5%

Table 1 shows the linear kernel SVM and RBF kernel SVM having the same accuracy and F-score of 58.2% and 68.5% respectively. This shows the close relations between the two SVM algorithms.

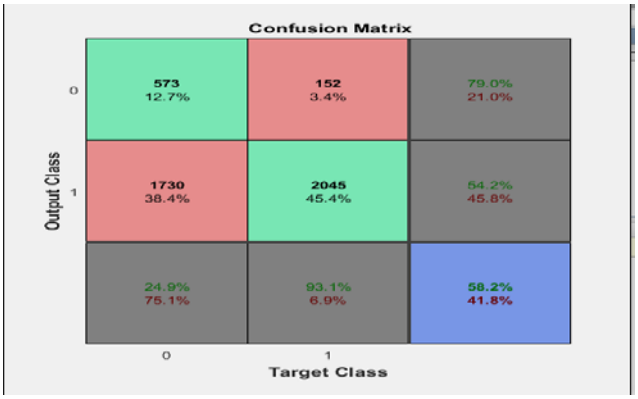

Fig.4. Confusion matrix of linear kernel SVM

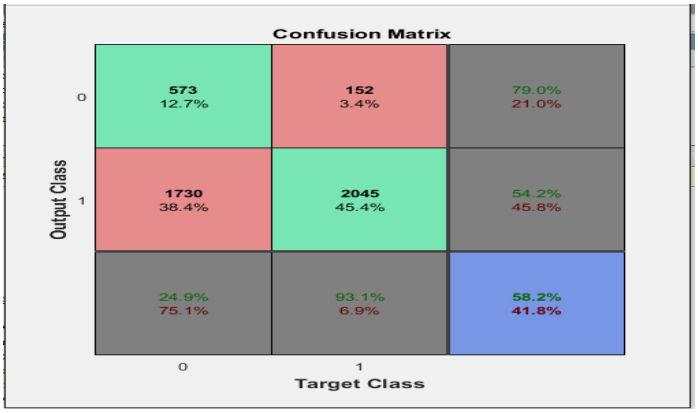

Fig.5. Confusion matrix of RBF kernel SVM

4.3. Training of decision tree model

The same training dataset was used to train a decision tree and the performance was evaluated using the test dataset. Figures 6 and 7 shows the plot of the resultant tree and the confusion matrix respectively.

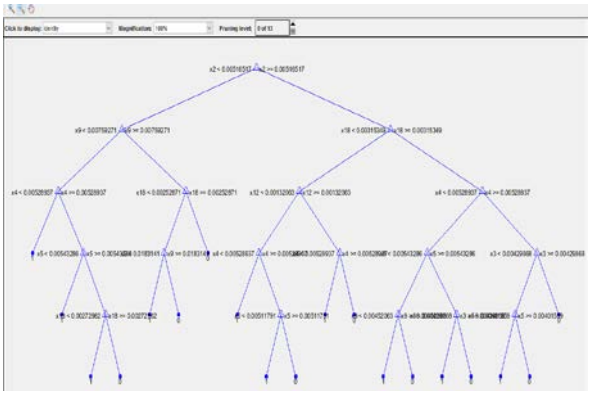

Fig.6. Plot of the trained decision tree

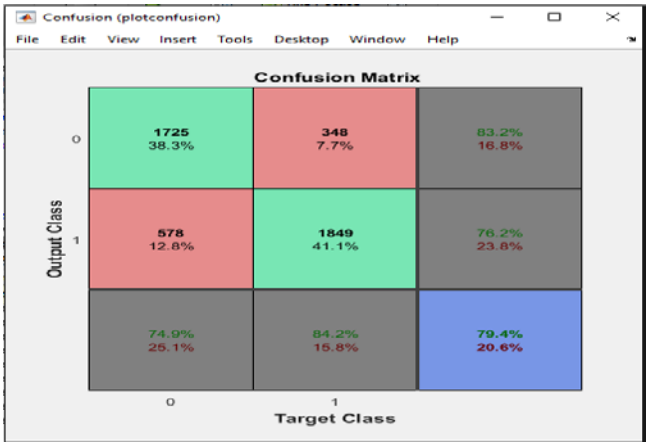

Fig.7. Confusion matrix of decision tree

Table 2. Performance result of decision tree

	Decision tree
Number of nodes	37
Accuracy	79.4%
Misclassification rate	20.6 %
Recall	76.2%
Specificity	83.2%
Precision	84.2%
F score	80%

The Table 2 shows the resultant performance of the decision tree giving 79.4% accuracy, 84.2% precision and F score of 80%, thus outperforming SVM which had an F score of 68.5% when tested on the same data set.

4.4. Training Artificial Neural Network (ANN)

A feedforward ANN with 10 hidden layers and a sigmoid activation function were trained using the same training dataset and the performance explored using the test dataset. Figure 8 and 9 shows the confusion matrix and receiver operating characteristics of the ANN model. The accuracies are tabulated in table 3.

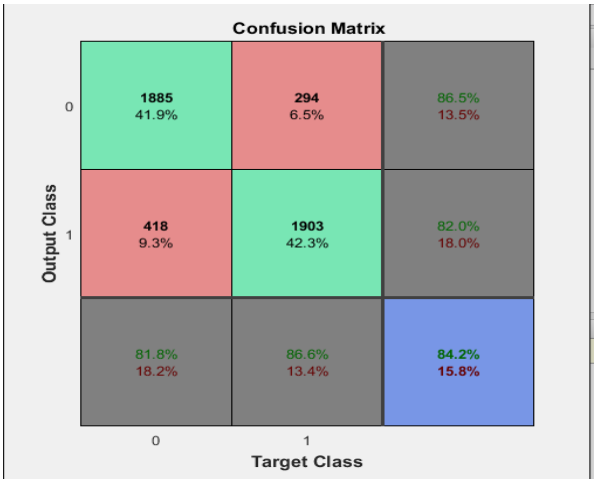

Fig.8. Confusion Matrix of ANN

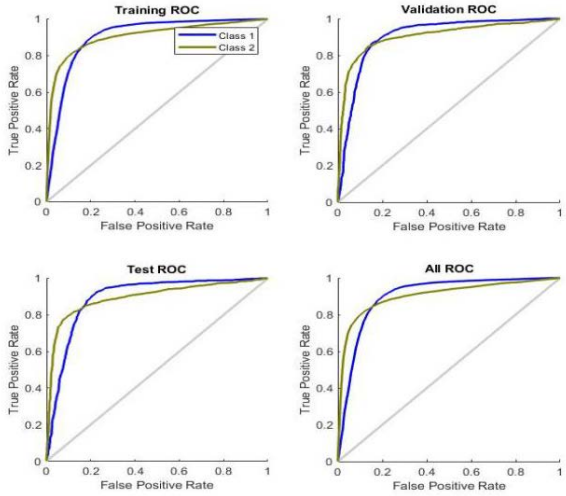

Fig.9. Receiver operating characteristics of ANN

Table 3. Performance result of ANN

	ANN
Number of hidden layer	10
Accuracy	84.2%
Misclassification rate	15.8%
Recall	82.0%
Specificity	86.5%
Precision	86.6%
F score	84.2%

From Table 3, when trained using the training set, ANN produced an accuracy of 84.2% and scored 84.2% using the F-score metrics.

4.5. Results of HCCFD

The confusion matrix of HCCFD is plotted as show in figure 10 and its accuracies are tabulated in table 4.

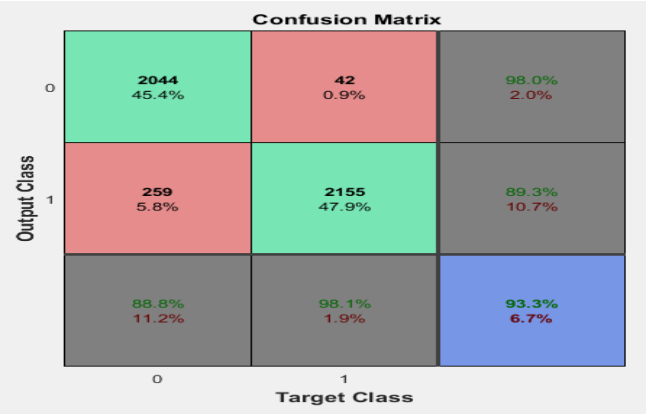

Fig.10. Confusion matrix of HCCFD

The results of our proposed model are shown in table 4. HCCFD showed an accuracy and F-score of 93.3% and 93.5% respectively; thus outperforming the other algorithms trained using the same data set.

Table 4. Performance result of HCCFD

	HCCFD
Accuracy	93.3%
Misclassification rate	6.7%
Recall	89.3%
Specificity	98.0%
Precision	98.1%
F score	93.5%

Table 5 gives a summary of our results in comparison with other widely used algorithms. HCCFD showed impressive results for all performance metrics applied, such as accuracy, miscalculation rate, recall, false positive rate, specificity, precision and F score. The F score of the hybrid model is 93.5% while support vector machine and decision tree are 68.5% and 80.0% respectively.

From all the results in our analysis, we have been able to prove that the new model outscores the existing models it was compared with, in terms of precision and specificity of results.

Table 5. Comparison of support vector machine, decision tree, ANN and HCCFD.

	Support Vector Machine	**Decision tree**	**ANN**	**HCCFD**
Accuracy	58.2%	79.4%	84.2%	93.3%
Misclassification Rate	41.8%	20.6%	15.8%	6.7%
Recall	54.2%	76.2%	82.0%	89.3%
False Positive Rate	21.0%	16.8%	13.5%	2.0%
Specificity	79.0%	83.2%	86.5%	98.0
Precision	93.1%	84.2%	86.6%	98.1
F Score	68.5%	80.0%	84.2%	93.5%

5. Conclusion

A hybrid model for detecting credit card fraud is developed using the general concept of outlier by applying genetic algorithm and multivariate normal distribution on an unbalanced credit card transaction dataset. The prediction accuracy of the model was compared with that of artificial neural network, support vector machine, and decision tree, after being trained and tested on the same dataset. The results obtained from the model gave an impressive F score of 93.5% while artificial neural network, support vector machine and decision tree obtained 68.5%, 84.2% and 80.0% respectively, as their F score.

In the past, efforts have been made to use some algorithms both individually and in combination with some others. Our research has proved that combining our statistical algorithm for outlier detection (multivariate normal distribution) with genetic algorithm yields a very high performance. This study reinforces the efficacy of multivariate normal distribution and genetic algorithm as a research tool and laid a solid ground work to be used in an operational fraud and other transaction anomaly detection systems. Future works should focus on experimenting our proposed approach on real-life fraud detection in banking solutions.

References

[1] Samaneh Sorournejad, Zahra Zojaji, Reza Ebrahimi Atani, and Amir Hassan Monadjemi (2016). A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective. *arXiv:1611.6439*

[2] Kehinde, James Sunday PhD. (2015). Banking Sector Technology Discrepancies: The Cost and Effect on Service Delivery. *European Journal of Business and Management*. ISSN 2222-1905 (Paper) ISSN 2222-2839 (Online) *Vol.7, No.7*

[3] V. Filippov, L. Mukhanov, and B. Shchukin (2008). Credit Card Fraud Detection System. *7th IEEE International Conference on Cybernetic Intelligent Systems*, 1-6, 2008

[4] Amanze, B.C., and Onukwugha, C.G (2018). Credit Card Fraud Detection System in Nigeria Banks Using Adaptive Data Mining and Intelligent Agents: A Review. *International journal of scientific & technology research*, 7(7).

[5] Bart, B., Veronique V. & Wouter, V. (2015). Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection. *John Wiley Sons Inc*

[6] Reurink, A. (2016). *Financial fraud: A literature review* (No. 16/5). MPIfG Discussion Paper.

[7] Yifu, D. & Isabel, G. (2017). Using Uber Engineering to Combat Fraud in Real Time. Retrieved November 15, 2017, from https://eng.uber.com/mastermind/

[8] Delamaire, L, Abdou, HAH and Pointon, J (2009). Credit card fraud and detection techniques: a review. *Banks and systems*, 4(2)

[9] Zareapoor, M., Seeja, K. R., & Alam, M. A. (2012). Analysis on Credit Card Fraud Detection Techniques: Based on Certain Design Criteria. *International Journal of Computer Applications*, *52*(3).

[10] Dal Pozzolo, A., & Bontempi, G. (2015). Adaptive machine learning for credit card fraud detection.
[11] Prakash, A., & Chandrasekar, C. (2013). A parameter optimized approach for improving Credit card fraud detection. *International Journal of Computer Science*, *10*.
[12] Jha, S., Guillen, M., & Westland, J. C. (2012). Employing transaction aggregation strategy to detect credit card fraud. *Expert systems with applications*, *39*(16), 12650-12657.
[13] MohdAvesh Z., Jabir D. & Ali Haider E. (2014). Credit Card Fraud Detection System Using Hidden Markov Model and K-Clustering. *International Journal of Advanced Research in Computer and Communication Engineering. Vol. 3.*
[14] Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, *2*(2), 88-105.
[15] O. M. Elzeki, M. F. Alrahmawy, Samir Elmougy, "A New Hybrid Genetic and Information Gain Algorithm for Imputing Missing Values in Cancer Genes Datasets", *International Journal of Intelligent Systems and Applications*, Vol.11, No.12, pp.20-33, 2019.
[16] Hamdy M. Mousa, "Bat-Genetic Encryption Technique", *International Journal of Intelligent Systems and Applications*, Vol.11, No.11, pp.1-15, 2019.
[17] Nasim Soltani Soulegan, Behrang Barekatain, Behzad Soleimani Neysiani, "MTC: Minimizing Time and Cost of Cloud Task Scheduling based on Customers and Providers Needs using Genetic Algorithm", *International Journal of Intelligent Systems and Applications*, Vol.13, No.2, pp.38-51, 2021.
[18] Behzad Soleimani Neysiani, Nasim Soltani, Reza Mofidi, Mohammad Hossein Nadimi-Shahraki, "Improving Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm", *International Journal of Information Technology and Computer Science*, Vol.11, No.2, pp.48-55, 2019.
[19] Jyoti S. Kulkarni, Rajankumar S. Bichkar, " Optimization in Image Fusion Using Genetic Algorithm", *International Journal of Image, Graphics and Signal Processing*, Vol.11, No.8, pp. 50-59, 2019.
[20] S. Gopa; Krishna Patro & Kishore Kumar Sahu (2015). Normalization: A Preprocessing Stage. arXiv:1503.06462v1[cs.OH]
[21] I. -C. Yeh & C. – H. Lien (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36 (2473-2480)
[22] Olufade F.W. Onifade, Joseph D. Akinyemi, Olashile S. Adebimpe,"A Recursive Binary Tree Method for Age Classification of Child Faces", *International Journal of Modern Education and Computer Science*, Vol.8, No.10, pp.56-66, 2016.

Authors' Profiles

Makolo, Angela obtained a PhD in computer science from the University of Ibadan with a specialization in Bioinformatics. She is currently a Senior Lecturer at the Computer Science department, University of Ibadan, Ibadan Nigeria. She has published over 40 papers in both local and international referred journals and conferences and has held several fellowships including ETT-MIT and the TechWomen Emerging Leader Fellowship. Her research interests imclude Computational Biology, Bioinformatics, Machine Learning and Software Engineering. Dr. Makolo is member of ISCB and CPN.

Adeboye I. Tayo obtained a Diploma in Statistics from Federal School of Statistics Kaduna, B.Tech in Cyber Security Science from Federal University of Technology Minna, and M.sc. in Computer Science from the University of Ibadan Nigeria, in 2010, 2016 and 2021 respectively. His research interests includes Information and Network Security, Machine Learning and Bioinformatics. Tayo is a member of the Cyber Security Experts Association of Nigeria (CSEAN), Nigeria Youth Internet Governance Forum (NYIGF), as well as a Digital Grassroots Ambassador.

How to cite this paper: Angela Makolo, Tayo Adeboye, "Credit Card Fraud Detection System Using Machine Learning", International Journal of Information Technology and Computer Science(IJITCS), Vol.13, No.4, pp.24-37, 2021. DOI: 10.5815/ijitcs.2021.04.03