AN EFFICIENT APPROACH FOR CREDIT CARD FRAUD DETECTION

Rajeev Kumar[1*]
Master of Computer Application
Department of Master of Computer Application
Jain Deemed-to-be university, Bangalore, India

Rajesh Budihul[2]
Master of Technology
Department of Master of Computer Application
Jain Deemed-to-be university, Bangalore, India

Abstract - The purpose of this research paper, the topic of credit card fraud detection has gained and developed fraudsters are increasing day by day among researches because of their frequent look in varied and widespread application within the field of various branches of information technology and engineering. For example, genetic algorithms, Behavior-based techniques, and Hidden Marks models are also used to address these problems of technology. Credit card fraud detection models for transactions are tested individually and proceed to whatever is most effective. This thesis aims to detect fraudulent transactions and develop some method of generating test data. These algorithms are a predictive approach in solving high complexity computational problems. We discussed a new method to goal or deal with detect fraud by filtering the above techniques to induce an improved result. These algorithms are a predictive approach in solving high complexity computational problems. It is an adaptation technique and evolutionary discovery that supports the existence of genetic and fittest. Implementation of efficient credit card fraud detection systems is mandatory for all credit card issuing companies or their customers to reduce their losses.

Keyword - Fraud detection of credit card; Naive Bayes Classifier; K-Nearest Neighbors Classifier; Logistic Regression Classifier; Bayesian learning; Hidden Markov Model; K-means Clustering; Group Method of data Handling model; Dempster Shafer Theory and Neural Network.

1. INTRODUCTION

A credit card is a thin working plastic card, identification information, such as a signature or photo [1] and authorizes the person named after him for the purchase fee or services to his account - fee for which he will bill from time to time. Today, there is data on the card automated teller machines (ATMs), read by store readers, also used in bank and online internet banking system. They have a unique card number which is extremely important. Its security depends on physical security plastic card and credit card secrets number [2]. Credit card is useful in our life from day by day. Our aim here is to detect fraud so that fraud can be detected or detected before fraud can occur. The goal is to minimize and accurately detect false fraud.

Credit card numbers have grown rapidly in transactions that have led to a sufficient increase in deceptive activity. Credit card fraud is a wide-ranging term of theft or fraud as the source of credit card fraud in a given transaction. Statistical methods and several data mining algorithms are commonly used to solve this fraud detection problem. Most credit card fraud detection systems are based on transaction behavior [3] using artificial intelligence, machine learning and data analysis.

In this paper, we will emphasis on credit card fraud and procedures to detect it. When credit card fraud occurs, the person uses the cards of other persons for their personal use without the information of their owner. When such cases are executed by fraudsters, they are used until it's completion and it is available limit reached.

Thus, we need a resolution that reduces the total limits available on credit cards that are more prominent for fraud. And, these model techniques or methods produce better solutions as time progresses. Full emphasis has been laid on developing accomplished and secure e-payment systems for fraud detection.

2. CREDIT CARD FRAUD METHODS

There are several ways to detect credit card fraud, as well as k-means clustering [4], hidden Markov models [5], grouping of data handling models [6], dempster Shafer theory [7], bayesian learning, and neural networks.

2.1. K-means Clustering methods

k-means clustering is basically a method of vector quantization from signal processing, which aims to divide n observations into k clusters, with each observation falling under a cluster with the closest mean (i.e., cluster centre or cluster centroid). Serves as a prototype. Cluster. The K-means k attempts to divide x data-points into sets of clusters, where each data-point is assigned to its nearest cluster. This method is defined by the objective function that tries to minimize the sum of all class distances within the cluster for all clusters [4, 8].

Given a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k \leq n$ sets S = $(s_1, s_2, \ldots, s_k)$ so as to minimize the within-cluster sum of squares.

Formally, the purpose of K-means clustering is to find: -

$$\Rightarrow arg_{S^{min}} \sum_{a=1}^{q} \left(\sum_{x_b \varepsilon S_a} \|x_b - \mu_a\|^2 \right)$$

$$\Rightarrow arg_{s^{min}} \sum_{i=1}^{k} |s_i| \mathrm{Var}\, s_i$$

where μ_i is the mean of points in s_i. This is equivalent to minimizing the pair-wise squared deviations of points in the same cluster: -

$$\Rightarrow arg_{s^{min}} \sum_{i=1}^{k} \frac{1}{2|s_i|} \sum_{x.y \in S_i} \|x - y\|^2$$

Or, validation can be reduced by recognition: -: -

$$\Rightarrow \sum_{x \in S_i} \|x - \mu_i\|^2$$

$$\Rightarrow \sum_{x \neq y \in S_i} (x - \mu_i)(\mu_i - y)$$

Steps	Disuses of k-means clustering algorithm
Step 1:	Select the value of q and number of clusters to be formed.
Step 2:	Randomly select q data-points from the dataset as the initial cluster centroids/centres.
Step 3:	For each data-point: - a. Compute the distance between the data-point and the cluster centroid b. State the data-point to closest centroid
Step 4:	For each cluster calculate the new mean based on the data-points in the cluster.
Step 5:	Quote step 3 & step 4 until mean of the clusters stops changing or maximum number of iterations reached.

Table 1

2.2. Hidden Markov Model methods

The hidden Markov model [9] (HMM) is a class of probabilistic graphical models that allows us to predict a sequence of unknown (hidden) variables from a set of observed variables. For example, of HMM is predicting (hidden variables) during the season that the fabric is based on one type of fabric. In the order of steps used to predict the best classification of hidden states, HMM can be watched as a bias net in one order of time.

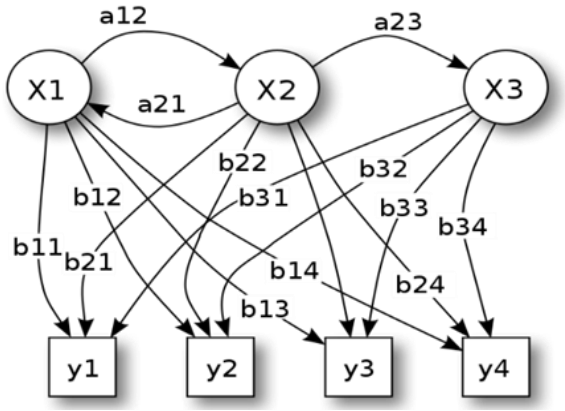

Figure 1: Probabilistic parameter of a hidden Markov model

Where, X = denoted by states of process
y = denoted by possible observations

a = denoted by state transition probabilities
b = dented by output probabilities

In figure 1, Wikipedia refers to an HMM and its transition. The script is a room containing urns x1, x2 and x3, each of which has a known mixture of balls, each ball labeled y1, y2, y3 and y4. A sequence of four balls is drawn randomly. In this particular case, the user looks at the order of balls y1, y2, y3 and y4 and is trying to understand the hidden position which is the correct sequence of the three urns that were drawn from these four balls [5].

2.3. Group Method of data Handling model

GMDH stands for Group Method of Data Handling. It is defined a family of inductive algorithms for computer-based mathematical modelling of multi-parametric datasets that facilitate fully automated structural and parametric optimization of models. GMDH is used in such areas as data mining, knowledge discovery, forecasting, complex system modelling, optimization, and pattern recognition. The Group Method of Data Handling algorithm is characterized by an inductive process that slowly sorts complex polynomial models and selects the best solution through external criteria. A GMDH model with multiple inputs and one output is a subset of the components of the base function [6].

Formally. Find the GMDH equation: -

$$Y(x_1, \ldots, x_m) = k_0 + \sum_{i=1}^{m} k_i \, fun_i$$

Where, fun is noted by elementary functions dependent on various sets of inputs,
k = coefficients and
m = number of the base function components.

2.4. Dempster Shafer Theory methods

DST stands for Dempster Shafer Theory. It is a general framework for reasoning with uncertainty makes sense with other frameworks such as probability and ineffective probability theories. Dempster Shafer theory [10] is constructed with two fundamental ideas: deriving the degree of belief for a question from subjective probabilities for the related question, and Dempster's rule of concatenate such parts of belief when they form an independent item of evidence be based on [7].

2.5. Bayesian learning methods

Bayesian learning requires a (possibly infinite) sum over the entire hypothesis space. Statistical learning approaches calculate the probability of each hypothesis ‘y’ given the data ‘x’, and select the hypothesis / make predictions based on this prediction makes predictions using all hypotheses weighted by their probabilities [11, 10].

Suppose, in the following: Set of fixed training $(t_1, t_2, \ldots, t_m)$ and classification of data $(x_1, x_2, \ldots, x_m)$ and determine the most likely hypothesis using the Bayes theorem.

$$P(y|x) = \frac{P(x|y)\, P(y)}{P(y)}$$

Where, P(y) = prior probability of hypothesis y
P(x) = prior probability of x
P(y|x) = probability of y given x
P(x|y) = probability of x given y

2.6. Neural Network methods

A neural network [12] is composed of an interconnecting array of processing units. The 'input nodes' are connected to one or more intermediate layers of nodes, called 'hidden units', which in turn feed to one or more output nodes. There may be more than one output layer. Each node in each layer is connected to all nodes in the previous and lower layers. Network processing capacity is stored in loads associated with interconnected units. Otherwise, a neural network is a simplified model of the way human brain processes. It works by simulating a large number of interconnected processing units that resemble abstract versions of neurons. Processing units are organized into layers. He arranges the processing units into layers. Processing units are organized into layers. He arranges the processing units into layers. A neural network consists of three parts: the first thing is an input layer, which consists of units representing input fields; one or more than the hidden layers; and an output layer with a unit or units reporting the target fields [13].

3. VARIOUS TECHNIQUES USED IN CREDIT CARD FRAUD

The advent of credit cards has not only given us privilege and convenience, but has also attracted malicious characters, as it is that slowing down is the best way to earn a relatively more money in a very short period of time. In addition, it takes ages to find out that the user has been cheated.

Some usual techniques used by the fraudster are: -

- Credit card copying and somehow capturing the user's secret PIN code.
- The credit card holder charges more cash to the user's credit card than they have to agree to listen later for the money charged.

So that not only the buyer, but also the credit card issuing bank, is at a loss and, therefore, has some interest to reduce illegality, that the use of credit cards is leading to the occurrence of various credit card fraud detection techniques. In order to detect credit card fraud, looking at a range of transactions and then identifying them and classifying them into critical transactions and thus implementing fraudulent transactions [1].

Fraud detection systems [14] are facing many difficulties and challenges. An effective fraud detection technique must have the ability to overcome these difficulties to achieve the best performance.

	Difficulties of credit card fraud detection.
Imbalanced data	Credit card fraud detection data have imbalanced nature. This means that a very small percentage of all credit card transactions are fraudulent. This makes detecting fraudulent transactions very difficult and obstructive.
Different misclassification importance	In the fraud detection task, different miscarriage errors have different significance. Diversification of a normal transaction as fraud is not as harmful as detecting a transaction as a normal fraud. Because the mistake in classification in the first case will be identified in further investigation.
Overlapping data	Many transactions can be considered fraudulent, whereas in reality they are normal (false positives) and vice versa, a fraudulent transaction may also appear to be valid (false negative). Therefore, achieving low rates of false

	positives and false negatives is an important challenge of fraud detection systems.
Lack of adaptability	Classification algorithms typically face the problem of detecting new types of common or fraudulent patterns. Supervision and obsolete fraud detection systems are inefficient in detecting new patterns of common and fraudulent behaviors, respectively.
Fraud detection cost	The system must take into account both the cost of fraudulent behavior and the cost of preventing it. For example, no revenue is obtained by stopping fraudulent transactions of a few dollars.

Table 2

4. LITERATURE REVIEW

In a point of review of the literature, in this paper [8] represents research regarding a case study involving credit card fraud detection, where data normalization is applied before cluster analysis and the importance of this paper on fraud. New methods and algorithms for detection were to be discovered and to extend the accuracy of the results. In this paper [15] predicts real-life transaction data by a European and had to find an algorithm that they found was the Bayes minimum risk. In this paper [16] we have found source code and how to process and find results. In this paper [17], with the help of python 3.0 console platform. This graph is showing the structure. In this paper gives a general description of the fraud detection systems developed during this fraud like various classifiers and therefore the model used different techniques and finally, conclusions are drawn about the results of the evaluation of the model.

Some major contributions to credit card fraud detection processes are discussed in Table 3 below.

Table 3: Literary review of credit card fraud detection procedures: -

Authors	Year	Handling various problems using credit card fraud detection methods
John Richard D. Kho and Larry A. Vea	1997	Authors proposed a method for a neural network-based using database mining system for credit card fraud detection.
Suvasini Panigrahi, Amlan Kundu, Shamik Sural and A. K. Majumdar	2009	Authors proposed a method for Credit card fraud detection of a fusion approach using Dempster – Shafer theory and Bayesian learning.
E. Aleskerov, B. fieisleben and B. Rao	2011	Authors proposed a method for credit card fraud detection using KNN, HMM and GMDH methods.
S. Benson Edwin Raj and S. Benson Edwin Raj	2017	Authors proposed a method for credit card fraud detection using various methods or discusses credit card fraud detection based on transaction behavior.

This table 4 discusses how credit card fraud is detected for various purposes.

Table 4: Literary review of credit card fraud detection methods: -

Authors	Year	Different approaches to solve these problems

Ishu Trivedi, Monika, Mrigya Mridushi	2013	They proposed algorithm for Performance analysis classification algorithm for data classification.
Khyati Chaudhary, Jyoti Yadav, Bhawna Mallick	2012	Author approaches a review of fraud detection techniques for credit card.
Tina R. Patil and Swati S. Sherekar.	2016	Author introduced the naive Bayes classifier; k-nearest neighbors classifier; logistic regression classifier and K-means Clustering.

The above discussions are that the problem of credit card fraud detection has gained various methods and techniques among researchers due to their consistent approach in diverse and wide-ranging applications and systems in the fields of various branches of science and engineering. Additionally this higher literature review suggests that research is for detecting credit card fraud within datasets derived from ULB by applying bayesian learning, hidden Markov model, k-means clustering, group method of data handling model, neural network, Dempster Shafer theory methods and various classifier applies as a naive Bayes, k-nearest Neighbors, logistic regression or random forest [18] and to estimate their accuracy, sensitivity, specificity, precision using various models and comparisons collide to tell them the simplest probabilistic model to settle the problem of credit card fraud detection.

5. DESCRIPTION OF EXISTING SYSTEM

Discussion for working on the Kaggle database, k-nearest neighbor (KNN) method, k-means clustering methods, hidden Markov model (HMM) methods, data handling model group method, demister shafar methods, Bayesian learning methods and Neural Network methods in case of existing system. Bayesian learning methods, neural network methods, and datasets from kaggle.com were collected and modified with a dataset of hybrid samples or Naive Bayes Classifier, K-nearest neighbor classifier, logistic regression classifier classified technology. To avoid the above-mentioned disadvantages throughout, we propose the existing system to detect fraud in a very good and direct way.

Table 5: Pros and Cons of credit card fraud detection shown in below: -

Sr. No.	Pros	Cons
01.	In the case of the existing system that even the first cardholder is additionally checked for fraud detection. But in these systems no have to check the first user as I maintain a log.	Indebtedness and Accrued fees are payable by the victim
02.	The log which is maintained will be proof for the bank for the transaction made.	Bad Credit Score and High-interest rates or annual fees related to credit cards
03.	I could find the foremost accurate detection using this system.	Consumers, use credit over ever before
04.	This reduces the tediously work of an employee within the bank.	High-cost fees

5.1. DESCRIPTION OF SURVEY WORK

Discussed the dataset utilized within the experiments or also three classifiers under study, namely; Naive Bayes, k-Nearest Neighbors, and Logistic Regression techniques. The assorted position involved in generating the classifiers include; a group of data, pre-processing of data, analysis of data, training of the classifier algorithm and testing (evaluation). These experiments are evaluated using True positive, True Negative, False Positive and False Negative rates metric. The performance comparison of the classifiers is analyzed supported accuracy, sensitivity, specificity, precision, Matthews parametric statistic, and balanced classification rate.

5.1.1. Dataset

The dataset is sourced from ULB Machine Learning Group [19] and description is found it. The dataset contains credit card transactions made by European cardholders in September 2013. This dataset presents transactions that occurred in two days, consisting of 284,807 transactions. The positive class (fraud cases) compose 0.172% of the transactions data [20]. The dataset is extremely unbalanced or skewed towards the positive class. It contains only numerical (continuous) input variables which are as a results of a Principal Component Analysis (PCA) feature selection transformation resulting in 28 principal components. Thus, an entire of 30 input features are utilized during this study. the most points or background information of the features cannot be presented due to confidentiality issues. The time feature contains the seconds elapsed between each transaction and so the primary transaction within the dataset. The 'amount' feature is that the transaction amount. These Feature 'class' is that the target class for the binary classification and it takes value 1 for positive case (fraud) and 0 for negative case (non-fraud) [21].

5.1.2. Hybrid sampling of dataset

Data is pre-processing and data that is distributed over it. A hybrid of under-sampling and over-sampling is distributed over a highly unbalanced dataset to realize two sets of distributions for analysis (10:90 and 34:64). This will be done by adding stepwise and subtraction of a data-point estimated between existing data-points until the over-fitting thread is reached [22].

$$PCA_{new} = \sum_{i=1}^{m} PCA + i \qquad \text{I}$$

$$NCA_{new} = \sum_{i=1}^{m} NCA - i \qquad \text{II}$$

$$m = mod\left(\frac{\left(\frac{NCA}{PCA}\right)}{2}\right) \qquad \text{III}$$

Where PCA_{new} = number of positive data-point,
NCA_{new} = number of negative data-point,
m = modulus of ratio,
(PCA/NCA) = quantity of positive or negative class datapoint in imbalanced dataset.

5.1.3. Naive Bayes classifier

Bayesian theory is supported by Naive Bayes or is a statistical approach, which supports selection as the best possible probability. Bayesian probability approximates unknown

probabilities from known values. It also allows prior knowledge and logic to be applied to uncertain details. This technique holds the assumption of conditional independence between features within the data. The Naive Bayes [23] classifier relies on the conditional probabilities (iv) and (v) of the binary classes (fraud and non-fraud) [22].

$$P(c_i|f_k) = \frac{P(f_k|c_i)*P(c_i)}{P(f_k)} \quad \text{IV}$$

$$P(f_k|c_i) = \prod_{i=1}^{m} P(f_k|c_i)\ k = 1,..,m; \quad i = 1,2 \quad \text{V}$$

where *m* is denoted by maximum number of features,

$P(f_k|c_i)$ = probability of feature value f_k being in class ci,

$P(f_k|c_i)$ = probability of generating feature value f_k given class c_i,

$P(c_i)$ / $P(f_k)$ = probability of occurrence of class c_i and probability of feature value f_k occurring.

The classifier performs the binary classification supported Bayesian classification rule.

$$If\ P(c_1|f_k) > p(c_2|f_k) then\ the\ classification\ is\ c_1$$

$$If\ P(c_1|f_k) > p(c_2|f_k) then\ the\ classification\ is\ c_2$$

where c_1 = negative class,

c_2 = positive class,

c_i = target class for classification.

5.1.4. K-Nearest Neighbors Classifier

K-Nearest Neighbor is ideal based learning that carries its classification, which supports measures of similarity, such as Euclidean, Manhattan, or Minkowski distance functions. The first two distance measures work with continuous variables while the third categorical variables. Euclidean distance measurements have been employed during this study for the KNN classifier [24]. The Euclidean distance (D_{ij}) between two input vectors (x_i, x_j) is given by:

$$D_{ij} = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2}\ k = 1,2,\dots\dots,m \quad \text{VI}$$

For every information within the dataset, the Euclidean distance between an input data-point and therefore the current point is calculated. The distances are sorted in increasing order and k items with lowest distances to the input data-point are selected. The majority class among their items is found it and thus the classifier returns the majority class due to the classification for the input point. Parameter tuning for, k is disbursed for k = 1, 3, 5, 7, 9, 11 and k = 3 showed optimal performance. Thus, the value of k = 3 is utilized within the classifier [22].

5.1.5. Logistic Regression Classifier

The logistic Regression which uses a functional approach to estimate the probability of a binary response supported one or more variables features. It finds the best-fit para the logistic Regression which uses a functional approach to estimate the probability of a binary response supported one or more variables features. It finds the best-fit parameters to a nonlinear function

called the sigmoid. The sigmoid function (σ) and so the input (x) to the sigmoid function is shown in (VII) and (VIII)

$$\sigma(x) = \frac{1}{(1+l^{-x})} \quad \text{VII}$$

$$x = w_0 z_0 + w_1 z_1 + \cdots + w_m z_m \quad \text{VIII}$$

The vector z is input file and also the simplest coefficients w, is multiplied together multiply each element and adds up to induce one number which determines the classifier classification of the target class. If the price of the sigmoid is over 0.5, it's considered a 1; otherwise, it's a 0. An optimization method is used to train the classifier and find the best-fit parameters. The gradient ascent (9) and modified stochastic gradient ascent optimization methods were experimented on to evaluate their performance on the classifier.

$$w := w + \alpha \nabla_w f(w) \quad \text{IX}$$

where the parameter ∇ is that the magnitude of movement of the gradient ascent. The steps are continued until a stopping criterion is met. The optimization methods are investigated (for iterations 50 - 1000) to understand if the parameters are converging. What are the parameters reaching a steady value or are they constantly changing? At 100 iterations, steady values of parameters are achieved.

The stochastic gradient updates the ascent incrementally as the new data comes in one go. It starts with all weights set to 1. Then for every feature value within the dataset, gradient ascent is calculated. The weight vector is updated by cargo of alpha and gradient. The load vector is then returned. Stochastic gradient ascent is used during this study because given the huge size of the information it updates the weights using only one instance at a time, thus reducing computational complexity [22].

5.2. SYSTEM DIAGRAM

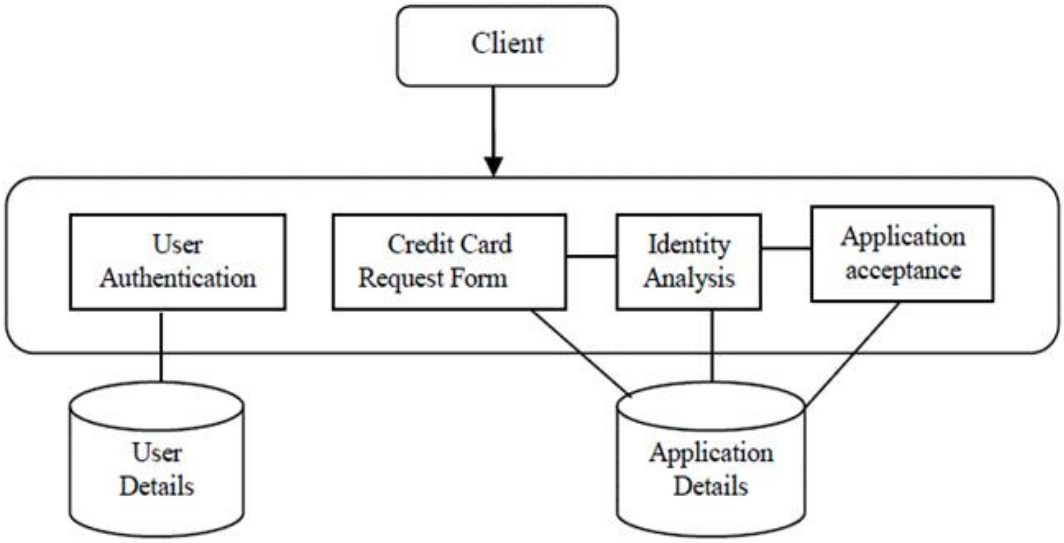

Figure 2: Architectural diagram [26]

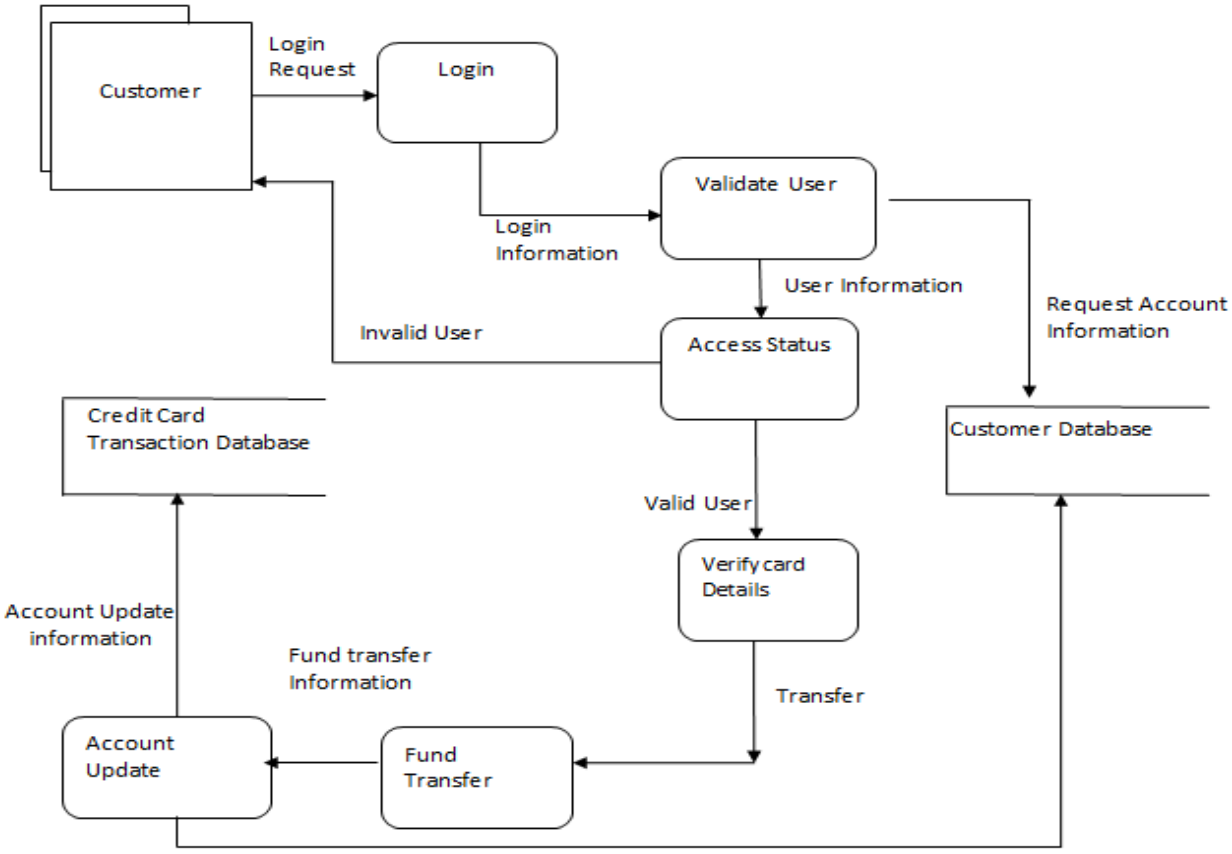

Figure3: DFD diagram [25]

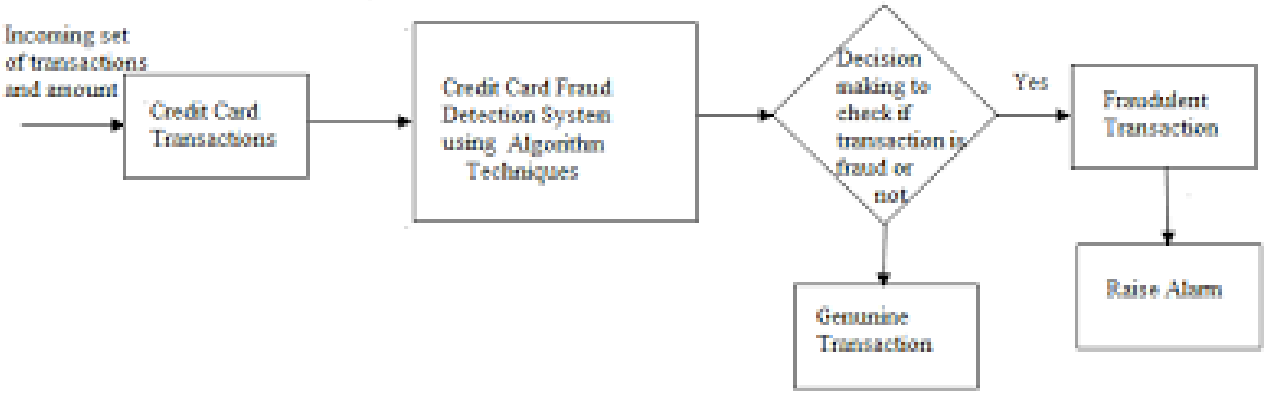

Figure 4: Block diagram [28]

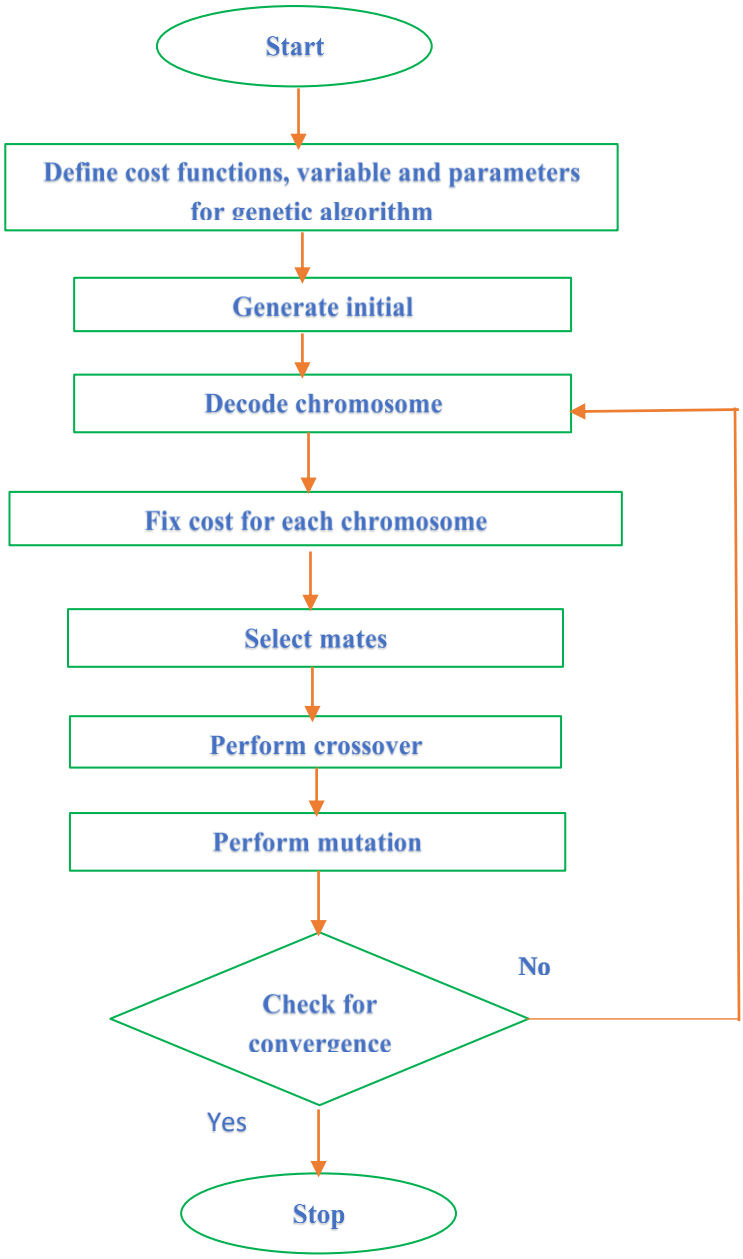

Figure 5: Process flow diagram [27]

6. PURPOSED SYSTEM

We have used a method, technique and an algorithm to calculate the probability of fraud of a credit card transaction. The algorithm outputs a classification (cheating / no cheating) and the probability of each, such as R (cheating) + R (no cheating) = 1. We want to rank the transaction so that not only can it be reviewed by the possibility of fraud, the amount at risk in each product.

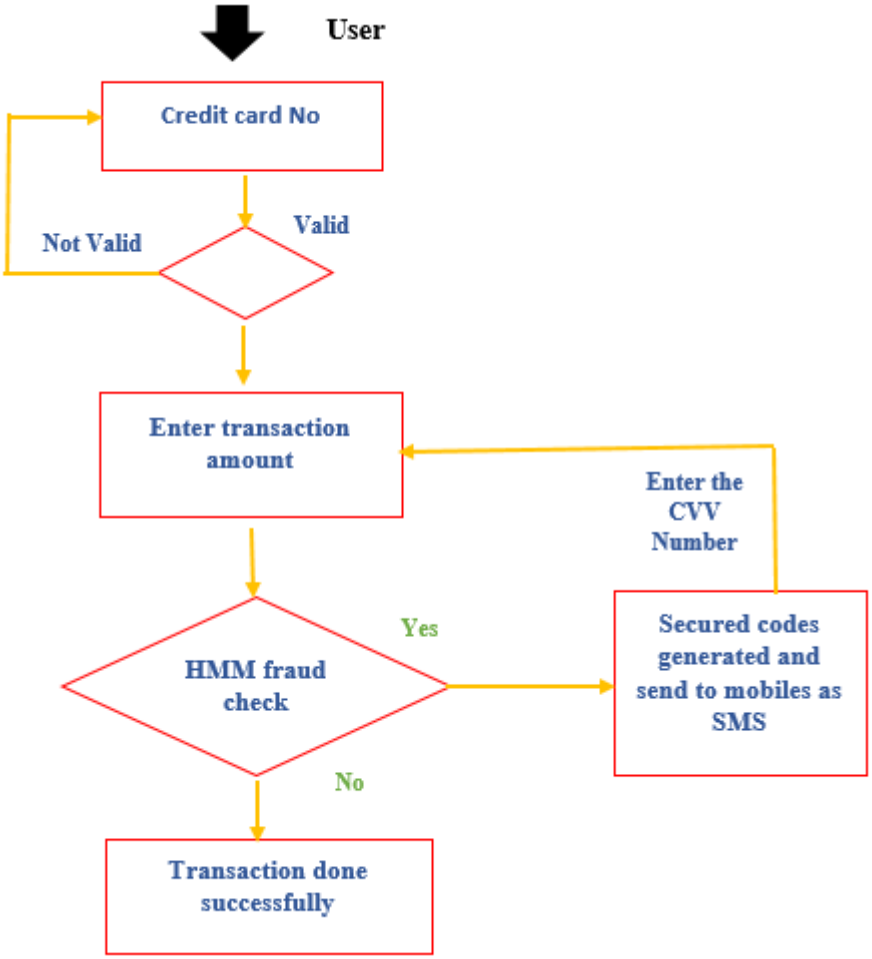

Figure 2: Flow chart of proposed system approaches for credit card fraud detection after training during detection.

6.1. The main challenges in purposed system are: -

- Hazardous data is processed day by day and models should be engineered apace to reply to scams in time.
- Unbalanced data i.e. most transactions (99.8%) don't seem to be fallacious that makes it very onerous for fraudsters to notice.
- Data availableness is usually personal within the type of knowledge.
- Misclassified knowledge is another major issue, as not each fallacious group action is caught and rumored.
- Adaptive technique employed by scammers against models

6.2. Purposed System solutions designed to deal these challenges: -

- The model used should be simple and fast to detect anomaly or quickly classify it as a fake transaction.
- Imbalance can be dealt with properly in a few ways which we will talk about in the next paragraph.
- Data mobility can be reduced to protect user privacy.
- A more reliable source must be taken that at least double-checks the data to train the model.

7. IMPLEMENTATION

We discuss building real-time solutions to detect credit card fraud. There are two steps to detect real-time fraud:

- The first step involves analysis or forensics on historical data to create a machine learning model.
- The second phase uses models in production to make forecasts for live events.

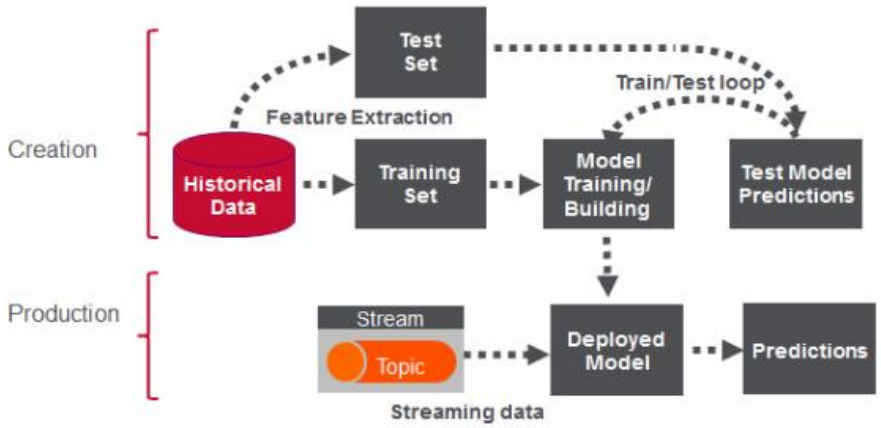

Figure 3: Evaluation system [29]

Exemplify the modelling of datasets using machine learning paradigm classification along with the basis detection of credit card fraud. Classification can also be a machine learning paradigm that involves obtaining a function that will separate the data into categories, or classes, with a training set of datasets (examples) of observations. This function is then employed to identify which categories the base observation is in.

7.1. Problem Statement

The credit card fraud problem involves modelling within the data of those in previous credit card transactions that turned out to be fraudulent. That model is then used to identify if a new transaction is bogus. The objective is to detect 100% of fraudulent transactions while reducing misclassification.

7.2. Solution methodologies

These datasets are collected at kaggle.com and analyzed during an exploratory collaboration with worldline and hence the machine learning group of ULB on big data mining and fraud detection. More data about current and past article on related topics is available at kaggle.com and hence the page of the credit card fraud investigation article. This dataset is picked up from kagle.com. We are used to the method of detecting credit card fraud.

Table 6: observations of dataset of credit card fraud detection is shown in below: -

Conditions	Observations
01.	These datasets have highly skewed, consisting amount of 492 frauds in a very total of 284,807 observations. This resulted in only 0.172% of fraud cases. This skewed set is justified by the low number of fraudulent transactions.

02.	These datasets carry with it numerical values from the 28 'Principal Component Analysis (PCA)' transformed features, namely V1 to V28. Furthermore, there's no metadata about the initial features provided, so pre-analysis or feature study couldn't be done.
03.	The 'Time' or 'Amount' features are not transformed data.
04.	It isn't a lost value within the dataset.

It is also seen that; a conclusion is drawn which is discussed below: -

- Such as an imbalanced data, a process that does not perform any kind of feature analysis and predicts all transactions, as non-fraud would also reach the target of accuracy of 99.828%. Therefore, accuracy is not an accurate measurement of efficacy in this case. We seek another standard of correctness, categorizing transactions as fraudulent or non-fraudulent.
- The 'special time' characteristic does not affect indicating the specific time of the transaction and is more than the list of information in sequential order. Therefore, we believe that the 'time' characteristic has less or no importance in the classification of fraudulent transactions. Therefore, we conclude this column by further analysis.

8. RESULT AND DISCUSSION

In this section; we are using data analysis to detect credit card fraud by ULB Machine Learning Group to provide fraud datasets and we have downloaded from the Kagle.com website. The dataset includes credit card transactions conducted by European cardholders in September 2013 [22].

This dataset represents transactions occurring over two days, where we found 492 fraud out of 284,807 transactions. The dataset is highly unbalanced, with positive squares (cheat) account for 0.172% of all classes. We used various forms of algorithm and sequence method and obtained the output with the result [21].

```
(284807, 31)
                Time            V1  ...        Amount          Class
count  284807.000000  2.848070e+05  ...  284807.000000  284807.000000
mean    94813.859575  3.919560e-15  ...      88.349619       0.001727
std     47488.145955  1.958696e+00  ...     250.120109       0.041527
min         0.000000 -5.640751e+01  ...       0.000000       0.000000
25%     54201.500000 -9.203734e-01  ...       5.600000       0.000000
50%     84692.000000  1.810880e-02  ...      22.000000       0.000000
75%    139320.500000  1.315642e+00  ...      77.165000       0.000000
max    172792.000000  2.454930e+00  ...   25691.160000       1.000000

[8 rows x 31 columns]
```

Figure 4: Describing the dataset

I described the data shape and print the dataset. We checked dataset in rows and columns format and calculate count, mean, std, min or max values and amount with different classes.

```
0.0017304750013189597
Fraud Cases: 492
Valid Transactions: 284315
```

Figure 5: Imbalance in the data

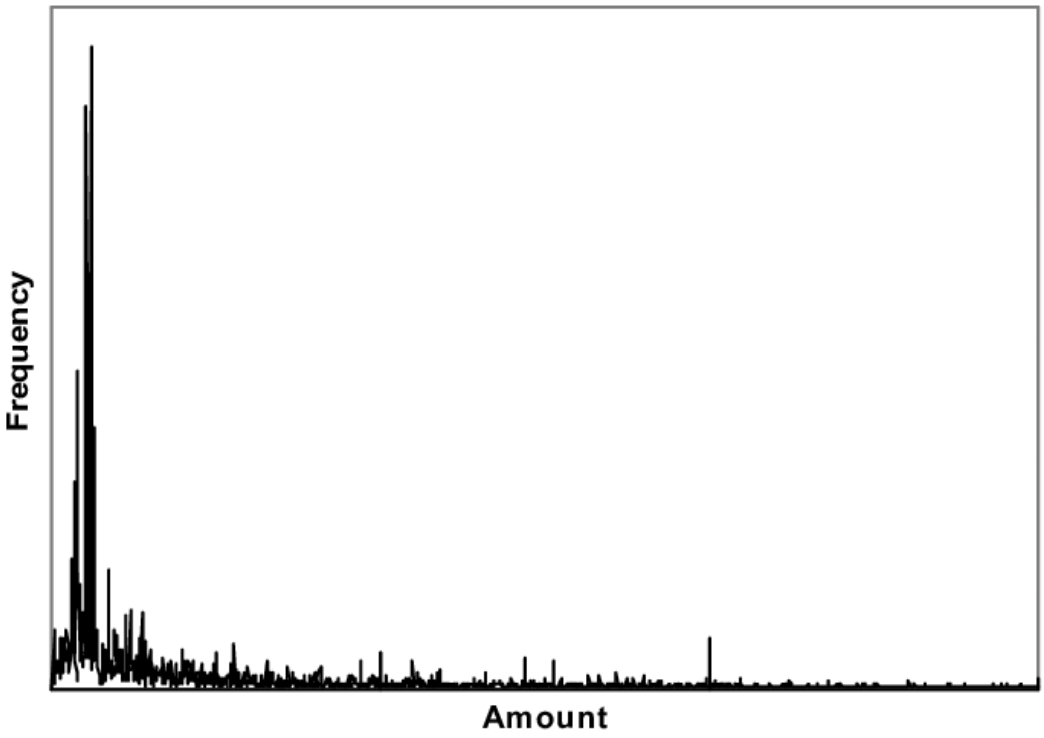

Figure 6: Histogram of amount vs frequency

Here, the amount of fraud cases within the dataset reflects and only 0.17% of fraudulent transactions are observed. Data is highly imbalanced. We investigated here, fraud cases 496 or legitimate transactions 284315.

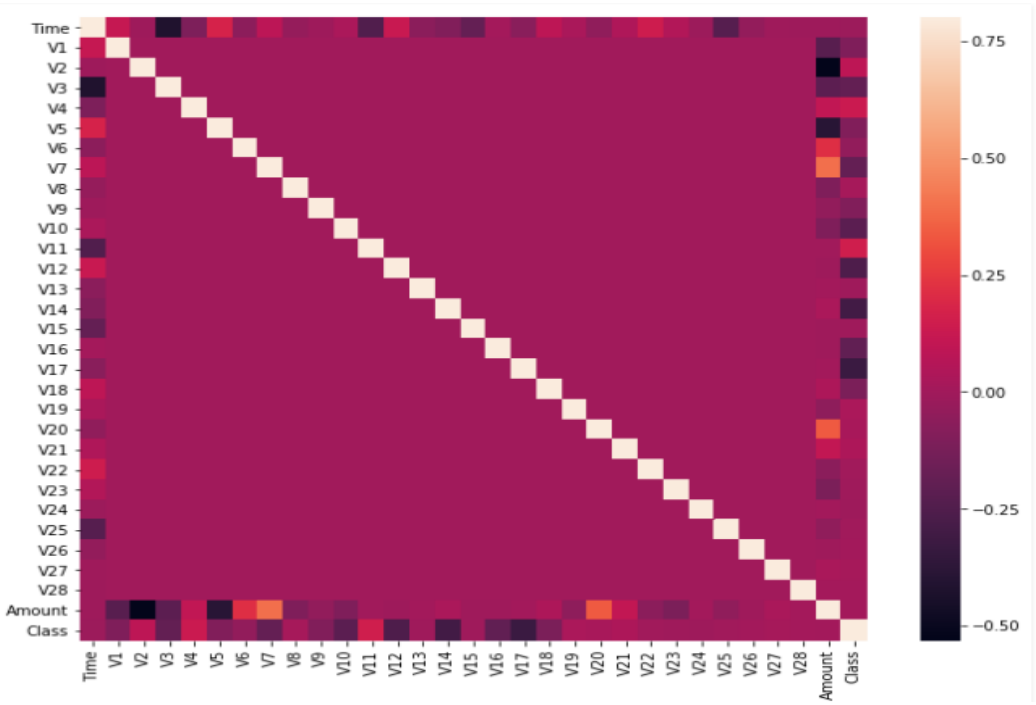

Figure 7: Plotting the Correlation Matrix

I am using to plot correlation matrix and I have checked the features v1, v2, ..., v28 class is compared to 'time' and 'amount'. In the heatmap; it is able to clearly see that almost all features are not associated with other features, but there are some features that involve either positive or negative correlation with each other. Here, v2 and v5 are highly negatively correlated with a feature called zodiac. I also see some connection with the v20 and the zodiac. This gives us a deeper understanding of the data available to us.

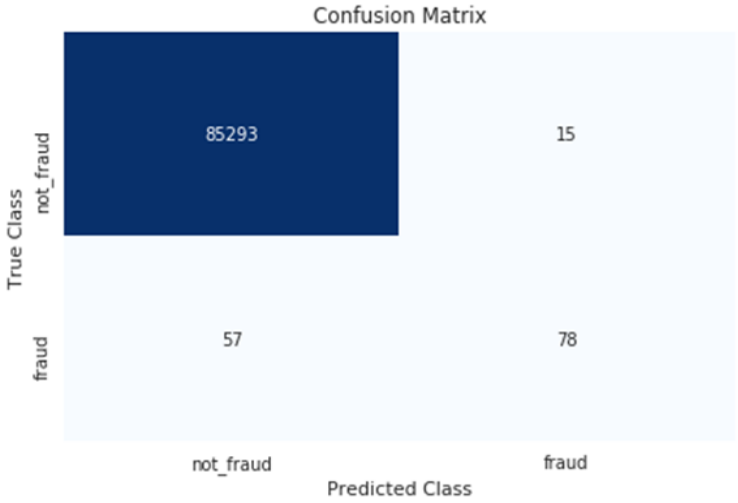

Figure 8: visualizing the confusion matrix filter_none

Here, Visualizing the Confusion Matrix. we printing the confusion matrix and labels(not_fraud or fraud) with comparing between true class and predicated class. We plotting plt.ylabel('True class') and plt.xlabel('Predicted class').

```
the Model used is Isolation Forest
The accuracy is  0.9978933323970366
The precision is 0.375
The recall is 0.336734693877551
The F1-Score is 0.3548387096774193
The Matthews correlation coefficient is0.3543008067850027
```

Figure 9: Find out the Matthews correlation coefficient value.

We evaluate the isolation forest or used model and train. The F1-score represents a more balanced result because it is the mean between precision and recall. We have found the Matthew correlation coefficient. We will apply various unbalanced data handling techniques and see their accuracy and miss the results. This result matches against the values of the class to check for false positives. Results when 10% of the dataset is used: -

```
Isolation Forest
Number of Errors: 71
Accuracy Score: 0.99750711000316

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.28      0.29      0.28        49

    accuracy                           1.00     28481
   macro avg       0.64      0.64      0.64     28481
weighted avg       1.00      1.00      1.00     28481


Local Outlier Factor
Number of Errors: 97
Accuracy Score: 0.9965942207085425

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.02      0.02      0.02        49

    accuracy                           1.00     28481
   macro avg       0.51      0.51      0.51     28481
weighted avg       1.00      1.00      1.00     28481
```

Figure 10: Find out the IF and LOC

Results with the complete dataset is used:

```
Isolation Forest
Number of Errors: 659
Accuracy Score: 0.9976861523768727

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    284315
           1       0.33      0.33      0.33       492

    accuracy                           1.00    284807
   macro avg       0.66      0.67      0.66    284807
weighted avg       1.00      1.00      1.00    284807


Local Outlier Factor
Number of Errors: 935
Accuracy Score: 0.9967170750718908

              precision    recall  f1-score   support

           0       1.00      1.00      1.00    284315
           1       0.05      0.05      0.05       492

    accuracy                           1.00    284807
   macro avg       0.52      0.52      0.52    284807
weighted avg       1.00      1.00      1.00    284807
```

Figure 11:IF or LOC values

9. PERFORMANCE EVOLUTION

Illustrations of the three classifications for the 34:66 data distribution in these demonstrations are shown in figure 16. These data distributions showed better performance. The k-nearest neighbor technique showed better performance in the evaluation matrix used for the two data distributions, a higher specificity and an accurate value of 1.0 were obtained. This may actually occur because the KNN classifier has not entered any false positives within the classification. The Naive Bayes classifier detected KNN inaccuracy for only 10:90 data distributions. Logistic regression classifier refers to the amount of performance between the three classifiers evaluated.

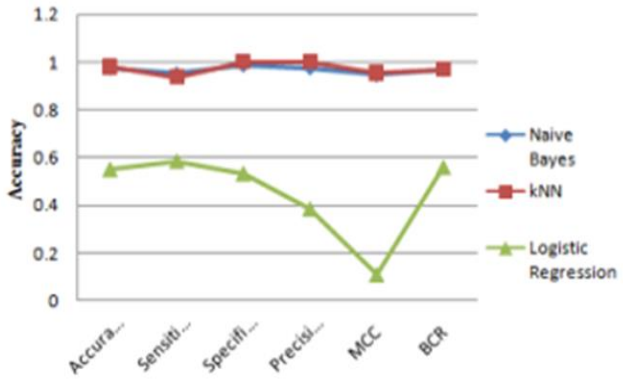

Figure 12: TPR and FPR evaluation chart for naive Bayes, KNN and logistic regression

However, there was a large improvement in performance between the two sets of sample data distributions. Since all related functions have not been evaluated with administered supported accuracy, sensitivity, specificity, accuracy, Matthew correlation coefficient, and balanced classification rate, this study compared other related functions with the required positive and false positive rates. Figures 17 and 18 propose Naive Bayes, KNN and LR classifiers against other related functions and are referred to in square brackets [ ].

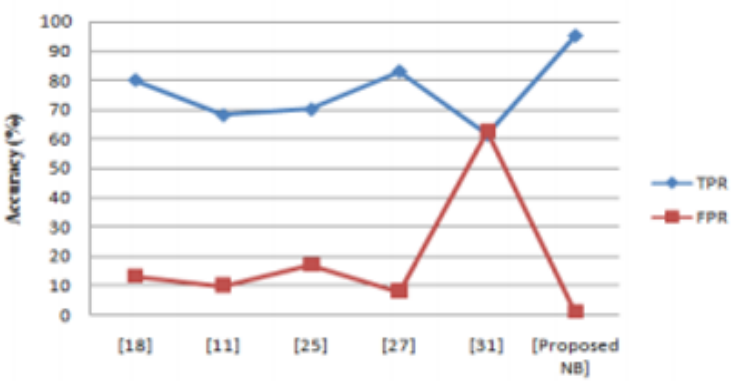

Figure 13: TPR and FPR evaluation of Naive Bayes classifiers

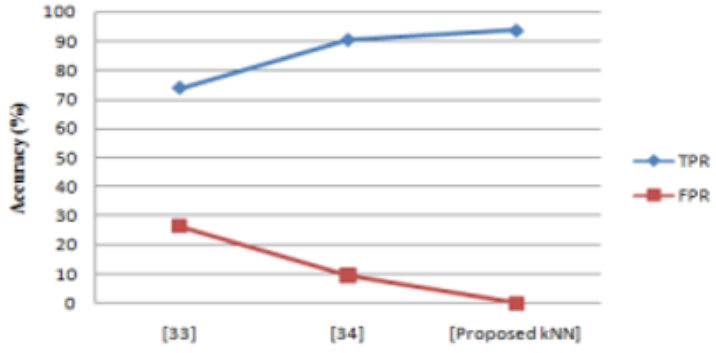

Figure 14: TPR and FPR evaluation of k-nearest neighbor classifiers

From this observation it is proposed that, KNN classifier recorded zero false positives for both sets of data distributions (i.e. 10:90 and 34:66 datasets) and the classifier compared the evaluation of positive and false positive rates on logistic to this time outperformed the reviewed works. The regression with other functions is shown in figure 19 and there is overlap between the true positive and false positive rates for the 10:90 data distribution as opposed to figures 17 and 18.

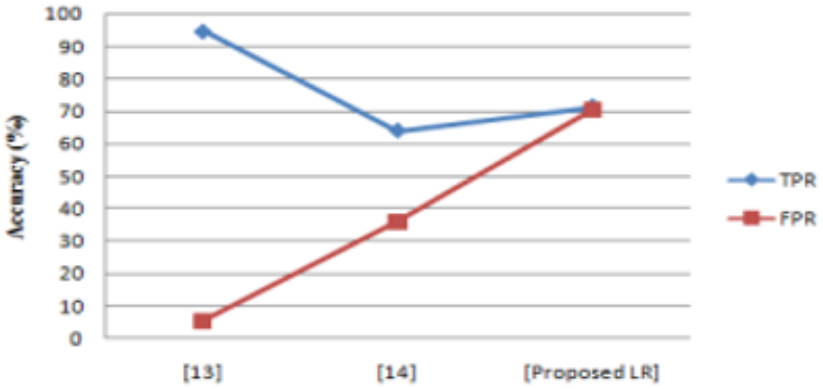

Figure 15: TPR and FPR evaluation of Logistic Regression classifiers

10. CONCLUSION

In this research study, we investigate the comparative performance of naive Bayes, k-nearest neighbor and logistic regression models within the binary classification of unbalanced credit card fraud datasets. The justification for examining these three techniques is due to their comparative ease as they are drawn to previous literature.

Final result, we perform classifiers differently in different evaluation metrics. Experiment results show that KNN shows significant performance for all matrices evaluated except accuracy within 10:90 data distribution.

The contribution of the paper is summarized within the following: -	
Contribution - 1	Three classifier-supported different machine learning techniques are trained over a critical lifetime of credit card transaction data and many relevant metrics comparing credit card fraud detection and their performance is supported.
Contribution - 2	Highly unbalanced dataset is measured in a highly hybrid approach, where the positive class is overlapped as well as the negative class is sampled, yielding sets of two data distributions.
Contribution - 3	Performance of three classifiers on sets of two data distributions is investigated using accuracy, sensitivity, specificity, precision, balanced classification rate, and Matthews statistical matrix.

Table 7

ACKNOWLEDGEMENT

I would like to express my gratitude and obligation to Professor Rajesh Budihul and Dr. M. N Nachappa for his effective conduct and constant motivations during his analysis work. Their timely direction, full cooperation and minute observation have made my work valuable. I would also like to

thank my mentor Professor Guru Basava, who wanted to provide me all the facilities that were required. Finally, I would like to thank my parents and friends for their support and encouragement throughout my studies.

REFERENCES

[1] I. Trivedi, Monika and M. Mridushi, "Credit card fraud detection," *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 5, no. 1, pp. 39--42, 2016.

[2] S. Vats, S. K. Dubey and N. K. Pandey, "A tool for effective detection of fraud in credit card system," *International Journal of Communication Network Security,* vol. 2, no. 1, 2013.

[3] J. R. D. Kho and L. A. Vea, "Credit card fraud detection based on transaction behavior," *TENCON 2017 - 2017 IEEE Region 10 Conference,* pp. 1880--1884, 2017.

[4] "k-means clustering," www.Wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering. [Accessed 13 April 2020].

[5] "Hidden Markov model," www.Wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Hidden_Markov_model. [Accessed 16 April 2020].

[6] "Grouping of data handling models," www.Wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Group_method_of_data_handling. [Accessed 26 March 2020].

[7] "Dempster–Shafer theory," www.Wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory. [Accessed 13 April 2020].

[8] A. Zafar and M. Sirshar, "A Survey on Application of Data Mining Techniques; It's Proficiency In Fraud Detection of Credit Card," *Research & Reviews: Journal of Engineering and Technology,* vol. 7, no. 1, pp. 15--23, 2016.

[9] A. Srivastava, A. Kundu, S. Sural and A. Majumdar, "Credit card fraud detection using hidden Markov model," *IEEE Transactions on dependable and secure computing,* vol. 5, no. 1, pp. 37--48, 2008.

[10] S. Panigrahia, A. Kundua, S. Surala and A.K.Majumdar, "Credit card fraud detection: A fusion approach using Dempster-Shafer theory and Bayesian learning," *Information Fusion*, vol. 10, no. 4, pp. 354--363, 2009.

[11] S. Maes, K. Tuyls, B. Vanschoenwinkel and B. Manderick, "Credit card fraud detection using Bayesian and neural networks," in *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, Brussel, Belgium, 2002.

[12] Ghosh and Reilly, "Credit card fraud detection with a neural-network," in *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, Wailea, HI, USA, USA, 1994.

[13] "Neural network," [Online]. Available: https://en.wikipedia.org/wiki/Neural_network. [Accessed 16 April 2020].

[14] S. Sorournejad, Z. Zojaji, R. E. Atani and A. H. Monadjemi, "A Survey of Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective," 2016.

[15] Suman and M. Bansal, "Survey paper on credit card fraud," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 3, no. 3, pp. 827--832, 2014.

[16] N. Demla and A. Aggarwal, "Credit card fraud detection using svm and reduction of false alarms," *International Journal of Innovations in Engineering and Technology*, vol. 7, no. 2, pp. 176--182, 2016.

[17] E. M. Carneiro, L. A. V. Dias, A. M. d. Cunha and L. F. S. Mialaret, "Cluster analysis and artificial neural networks: A case study in credit card fraud detection," in *12th International Conference on Information Technology-New Generations*, 2015.

[18] S. B. E. Raj and A. A. Portia, "Analysis on Credit Card Fraud Detection Methods," in *International Conference on Computer, Communication and Electrical Technology*, Coimbatore, 2011.

[19] L. Frei, "Detecting Credit Card Fraud Using Machine Learning," 2019.

[20] A. D. Pozzolo, O. Caelen, R. A. Johnson and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*, 2015.

[21] "Credit card fraud detection," Machine Learning Group- ULB, 23 March 2018. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud.

[22] J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics*, Lagos, Nigeria, 2017.

[23] T. R. Patil and S. S. Sherekar, "Performance comparison of naive bayes an J48 classification algorithms," *International Journal of Applied Engineering Research*, vol. 6, no. 2, pp. 256--261, 2013.

[24] M. J. Islam, Q. M. J. Wu, M. Ahmadi and M. A. Sid-Ahmed, "Investigating the performance of naive-bayes classifiers and k-nearest neighbor classifiers," in *International Conference on Convergence Informationd Technology*, 2007.

[25] "Data flow daigram of credit card fraud model," [Online]. Available:

https://images.app.goo.gl/xizK3ovrwwk7H1SZ7.

[26] "architectural diagram of credit card fraud system," [Online]. Available: https://images.app.goo.gl/mb5pUbXauDVP7CuEA.

[27] "Process flow diagram of credit card fraud detection," [Online]. Available: https://images.app.goo.gl/xHrAbdgcmaBQKcDW6.

[28] "Block diagram of credit card fraud model," [Online]. Available: https://images.app.goo.gl/zPKR751mQz7pBW8s6.

[29] "Evaluation system of credit card fraud detection," ULB-Machine Learning, [Online]. Available: https://images.app.goo.gl/hBo7NTVsHjqgGo4N7.