© 2018 IJSRSET | Volume 4 | Issue 10 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Discovery of Fraud in Credit Card by Combining Classifier and Clustering Machine Learning Techniques

Anette Regina. I*[1], Pavithra. J[2]

[1]Associate Professor, PG & Research Department of Computer Science and Applications, Muthurangam Government Arts College, Vellore, TamilNadu, India

[2]M.Phil (CS) Research Scholar PG & Research Department of Computer Science and Applications, Muthurangam Government Arts College, Vellore, TamilNadu, India

ABSTRACT

Because of the ascent and quick development of E-Commerce, utilization of credit cards for online buys has significantly expanded and it caused a blast in the charge card fraud. As credit card turns into the most prevalent method of installment for both online and general buy, instances of fraud related with it are additionally rising. Data mining system is one remarkable techniques utilized as a part of taking care of credit fraud detection problem issue. Credit card fraud detection is the way toward recognizing those exchanges that are deceitful into two classes of legitimate (certified) and fake exchanges. In actuality, fraudulent transactions are scattered with genuine exchanges and straightforward example coordinating procedures are not frequently adequate to recognize those fakes precisely. Usage of proficient extortion location frameworks has hence turned out to be basic for all credit card issuing banks to limit their misfortunes. The most ordinarily utilized misrepresentation recognition strategies are SVM calculations and Naïve Bayes. These systems can be utilized alone or in collaboration using ensemble or machine learning strategies to construct classifiers or Clustering method. This paper shows a review of different procedures utilized as a part of credit card fraud detection and assesses every approach in light of certain outline criteria. In this paper, clustering approach is introduced for classify the samples into several categories in credit card fraud detection. Information is produced arbitrarily for credit card and after that K- means clustering calculation is utilized for recognizing the transaction whether it is misrepresentation or real. Clusters are framed to recognize transaction exchange which are low, high, dangerous and high unsafe. After applying Clustering introduced naïve bayes and SVM on highly skewed credit card fraud data. The two techniques are applied on the raw and preprocessed data. The work is implemented in C#. The performance of the techniques is evaluated based on accuracy, sensitivity, specificity, precision, Matthews's correlation coefficient and balanced classification rate. The results shows of optimal accuracy for naïve bayes, SVM classifiers are 64.66%, 91.31% respectively. The comparative results show that SVM performs better than naïve bayes techniques.

Keywords : Credit Card, Fraud Detection, Data Generation, Anomalies, Machine Learning, K-Means Clustering Algorithm

I. INTRODUCTION

A. General Study

As credit card transactions become the foremost prevailing mode of payment for both on-line and offline transaction, credit card fraud rate additionally accelerates. Credit card fraud will come in either inner card fraud or external card fraud. Inner card fraud happens as a result of consent between cardholders and bank by using false identity to commit fraud whereas the external card fraud involves the use of stolen credit card to urge money through dubious means that. A lot of researches are dedicated to detection of external card fraud that accounts for majority of credit card frauds. Detection fraudulent transactions using traditional ways of manual detection is time overwhelming and inefficient, so the advent of huge information has created manual ways additional impractical. Data mining technique is one notable ways employed in finding credit fraud detection problem. Clustering has the appliance within the field of engineering and scientific disciplines like psychology, biology, medicine, pc vision, communication and remote sensing. a collection of pattern is determined by abstracting underlying structure in clustering. The patterns are clustered on the idea of additional similar options than alternative pattern of cluster. Different clustering algorithms are projected to fulfil totally different needs. Clustering algorithms are supported the structure of abstraction and are classified into hierarchic and partitioned algorithms. Hierarchic clustering algorithms construct a hierarchy of partitions that are depicted as a dendrogram during which every partition is nested within the partition at subsequent level within the hierarchy. Partitioned clustering algorithms, with a given or estimated range of non-overlapping clusters construct one partition of the information in an attempt to recover natural groups that are given within the data

II. LITERATURE SURVEY

Fraud detection involves watching the behavior of users so as to estimate, detect, or avoid undesirable behavior. To counter the credit card fraud effectively, it's necessary to know the technologies concerned in detecting credit card frauds and to spot various kinds of credit card frauds [1]. There are multiple algorithms for credit card fraud detection [2]. Artificial neural-network models that are primarily based} upon artificial intelligence and machine learning approach Meta learning Agents and Fuzzy based systems [3]. The opposite technologies concerned in credit card fraud detection are internet Services-Based cooperative scheme for credit card Fraud Detection within which participant banks will share the data regarding fraud patterns during a heterogeneous and distributed environment to reinforce their fraud detection capability and reduce loss [4].

A. Fusion approach by Dempster–Shafer theory and Bayesian learning

FDS of Dempster–Shafer theory and Bayesian learning Dempster–Shafer theory and Bayesian learning could be a hybrid approach for credit card fraud detection [5] which mixes evidences from current as well as past behavior. each cardholder contains a certain kind of shopping behavior, that establishes an activity profile for them. This approach proposes a fraud detection system by use of info fusion and Bayesian learning of therefore on counter credit card fraud. The FDS system consists of 4 elements, namely, rule-based filter, Dempster–Shafer adder, transaction history information and Bayesian learner. within the rule-based element, the suspicion level of every incoming transaction by the extent of its deviation from smart pattern is set. Dempster–Shafer's theory is employed to mix multiple such evidences and an initial belief is computed. Then the initial belief values are combined to get an overall belief by applying Dempster–Shafer theory. The transaction is classed as suspicious or suspicious based on this first belief. Once a transaction is found to be suspicious, belief is stronger or weakened consistent with its similarity with fraudulent or real transaction history by Bayesian learning.

B. Hidden Markov Model

A Hidden Markov Model may be a double embedded stochastic process with accustomed model much more difficult stochastic processes as compared to a standard Markov model. If an incoming credit card transaction isn't accepted by the trained Hidden Markov Model with sufficiently high likelihood, it's thought of to be fraudulent transactions. A Hidden markov Model [6] is at first trained with the traditional behavior of a cardholder. Every incoming transaction is submitted to the FDS for verification. FDS receives the card details and also the worth of purchase to verify whether or not the transaction is real or not. If the FDS confirms the transaction to be malicious, it raises an alarm and also the issue bank declines the dealing. The involved cardholder could then be contacted and alerted concerning the likelihood that the card is compromised.

C. The Evolutionary-Fuzzy System

Fuzzy Darwinian Detection system [3] uses genetic programming to evolve fuzzy logic rules capable of classifying credit card transactions into "suspicious" and "non-suspicious" categories. It describes the employment of an evolutionary-fuzzy system capable of classifying suspicious and non-suspicious credit card transactions. The system includes of a Genetic Programming (GP) search algorithm and a fuzzy professional system. Information is provided to the FDS system. The system 1st clusters the information into 3 teams particularly low, medium and high. The genotypes and phenotypes of the GP System comprises rules that match the incoming sequence with the past sequence. Genetic Programming is employed to evolve a series of variable-length fuzzy rules that characterize the variations between categories of information control in a information. The system is being developed with the precise aim of insurance-fraud detection that involves the difficult task of classifying information into the categories. "safe" and "suspicious". Once the customer's payment isn't due or the quantity of due payment is a smaller amount than 3 months, the transaction is taken into account as "non suspicious", otherwise it's thought-about as "suspicious". The Fuzzy Darwinian detects suspicious and non -suspicious information and it simply detects stolen credit card Frauds.

III. ANALYSIS AND DESIGN

A. Proposed System

Data mining approaches have been widely used for classification and prediction problems. The proposed approach is based on data mining, which consists of the K-means clustering and Naïve and KNN Classifiers. Fig. 1 shows the Credit Card Fraud Detection model. Firstly, data preprocessing is conducted on the vector space to clean unreasonable data, normalize the training samples and select the most related variables as the inputs of the Classifiers. Secondly, data after preprocessing are clustered by the K-means clustering to select the training set which is most similar to the transaction. Finally, Credit card Fraud is forecasted by the Naïve bayes and KNN Clustering.

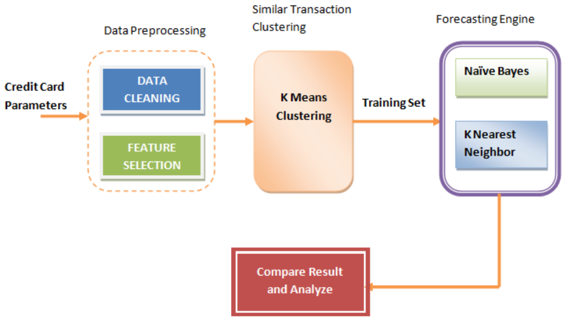


Figure 1. Credit Card Fraud Detection model

a) Data preprocessing

A number of Credit Card parameters are collected as the training samples via the sensor unit. However, these samples may contain unreasonable data. Besides, using too many parameters as the training features would increase the computing complexity and obtain undesired results for the reason that some variables are irrelevant or redundant in this model. Selecting features which are most related to the Credit card is able to improve the accuracy. Finally, data normalization has an effect on the convergence rate and accuracy of the training algorithm. Thus, in order to obtain accurate forecasting results, data preprocessing is necessary.

In this proposed approach clustering algorithm is utilized for credit card fraud detection. Information is produced arbitrarily for credit card and after that K-means clustering calculation is utilized for recognizing the transaction whether it is misrepresentation or real. Clusters are framed to recognize detect fraud in credit card transaction exchange which are low, high, dangerous and high unsafe. K-implies bunching calculation is straightforward and productive calculation for charge credit card fraud detection.

B. Algorithm

a) K-MEANS clustering Algorithm

K-MEANS have a tendency to outline center points and define K as number of clusters and it will observe noise and outlier by measure distance excellent, we will realize and optimize center point (here named Centroid) by repetition and rerunning the algorithm once more on the results of previous execution. So, the problem of this algorithmic rule is finding optimum K. We can sum up K-MEANS steps as

1. Input . K, number of clusters and n, objects dataset
2. Output. set of K cluster with minimum squared errors condition

Algorithm steps.

1) Take a number (K) of cluster centers – centroids (at random)
2) Assign each item to its nearest cluster center point
3) Move every cluster centre to the mean of its assigned items
4) Repeat the steps 2,3 until convergence (change cluster assignments less than the value of threshold)

b) Naive Bayes Algorithm

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below.

$$\underset{\text{Posterior Probability}}{P(c\mid x)} = \frac{\overset{\text{Likelihood}}{P(x\mid c)}\,\overset{\text{Class Prior Probability}}{P(c)}}{\underset{\text{Predictor Prior Probability}}{P(x)}}$$

$$P(c\mid X) = P(x_1\mid c)\times P(x_2\mid c)\times\cdots\times P(x_n\mid c)\times P(c)$$

Above,

- *P*(*c*/*x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*/*c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

c) SUPPORT VECTOR MACHINES

Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

In this post you will discover the Support Vector Machine (SVM) machine learning algorithm. After reading this post you will know.

- How to disentangle the many names used to refer to support vector machines.
- The representation used by SVM when the model is actually stored on disk.
- How a learned SVM model representation can be used to make predictions for new data.
- How to learn an SVM model from training data.
- How to best prepare your data for the SVM algorithm.
- Where you might look to get more information on SVM.

Maximal-Margin Classifier

The Maximal-Margin Classifier is a hypothetical classifier that best explains how SVM works in practice.

The numeric input variables (x) in your data (the columns) form an n-dimensional space. For example, if you had two input variables, this would form a two-dimensional space.

A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. For example.

B0 + (B1 * X1) + (B2 * X2) = 0

Where the coefficients (B1 and B2) that determine the slope of the line and the intercept (B0) are found by the learning algorithm, and X1 and X2 are the two input variables.

You can make classifications using this line. By plugging in input values into the line equation, you can calculate whether a new point is above or below the line.

- Above the line, the equation returns a value greater than 0 and the point belongs to the first class (class 0).
- Below the line, the equation returns a value less than 0 and the point belongs to the second class (class 1).
- A value close to the line returns a value close to zero and the point may be difficult to classify.
- If the magnitude of the value is large, the model may have more confidence in the prediction.

Support Vector Machines (Kernels)

The SVM algorithm is implemented in practice using a kernel.

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra, which is out of the scope of this introduction to SVM.

A powerful insight is that the linear SVM can be rephrased using the inner product of any two given observations, rather than the observations themselves. The inner product between two vectors is the sum of the multiplication of each pair of input values.

For example, the inner product of the vectors [2, 3] and [5, 6] is 2*5 + 3*6 or 28.

The equation for making a prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows.

f(x) = B0 + sum(ai * (x,xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

Linear Kernel SVM

The dot-product is called the kernel and can be re-written as.

K(x, xi) = sum(x * xi)

The kernel defines the similarity or a distance measure between new data and the support vectors. The dot product is the similarity measure used for linear SVM or a linear kernel because the distance is a linear combination of the inputs.

Other kernels can be used that transform the input space into higher dimensions such as a Polynomial Kernel and a Radial Kernel. This is called the Kernel Trick.

It is desirable to use more complex kernels as it allows lines to separate the classes that are curved or even more complex. This in turn can lead to more accurate classifiers.

Polynomial Kernel SVM

Instead of the dot-product, we can use a polynomial kernel, for example.

K(x,xi) = 1 + sum(x * xi)^d

Where the degree of the polynomial must be specified by hand to the learning algorithm. When d=1 this is the same as the linear kernel. The polynomial kernel allows for curved lines in the input space.

Radial Kernel SVM

Finally, we can also have a more complex radial kernel. For example.

K(x,xi) = exp(-gamma * sum((x – xi^2))

Where gamma is a parameter that must be specified to the learning algorithm. A good default value for gamma is 0.1, where gamma is often 0 < gamma < 1. The radial kernel is very local and can create complex regions within the feature space, like closed polygons in two-dimensional space.

How to Learn a SVM Model

The SVM model needs to be solved using an optimization procedure.

You can use a numerical optimization procedure to search for the coefficients of the hyperplane. This is inefficient and is not the approach used in widely used SVM implementations like LIBSVM. If implementing the algorithm as an exercise, you could use stochastic gradient descent.

There are specialized optimization procedures that re-formulate the optimization problem to be a Quadratic Programming problem. The most popular method for fitting SVM is the Sequential Minimal Optimization (SMO) method that is very efficient. It breaks the problem down into sub-problems that can be solved analytically (by calculating) rather than numerically (by searching or optimizing).

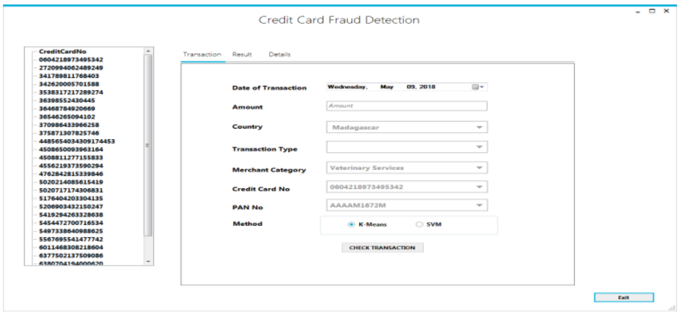
Data Preparation for SVM

This section lists some suggestions for how to best prepare your training data when learning an SVM model.

- Numerical Inputs. SVM assumes that your inputs are numeric. If you have categorical inputs you may need to covert them to binary dummy variables (one variable for each category).
- Binary Classification. Basic SVM as described in this post is intended for binary (two-class) classification problems. Although, extensions have been developed for regression and multi-class classification.

IV. RESULT

A. Screenshots

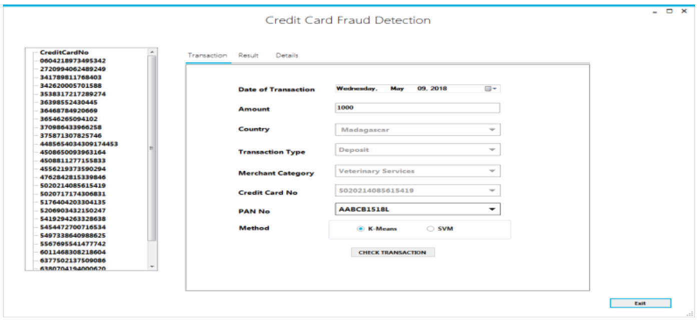


Credit Card Fraud Detection

CreditCardNo
0604218973495342
2720994062489249
341789811768403
342620005701588
3538317217289274
36398552430445
36468784920669
36546265094102
370986433966258
375871307825746
4485654034309174453
4508650093963164
4508811277155833
4556219373590294
4762842815339846
5020214085615419
5020717174306831
5176404203304135
5206903432150247
5419294263328638
5454472700716534
5497338640988625
5567695541477742
6011468308218604
6377502137509086
6380704194000620

Date of Transaction: Wednesday, May 09, 2018
Amount: Amount
Country: Madagascar
Transaction Type:
Merchant Category: Veterinary Services
Credit Card No: 0604218973495342
PAN No: AAAAM1672M
Method: ● K-Means ○ SVM

Figure 2. Load credit card transactions

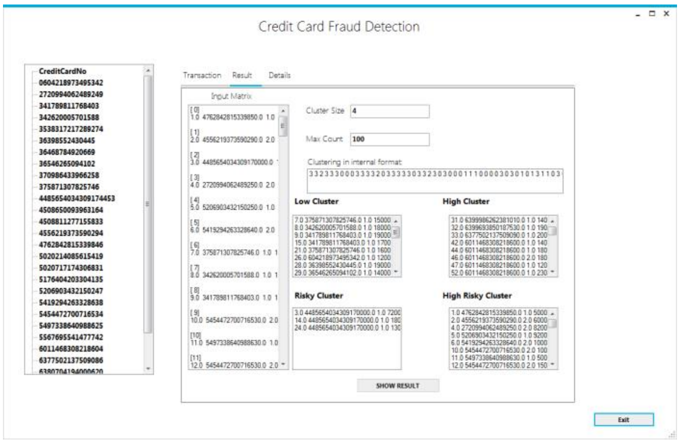


Credit Card Fraud Detection

CreditCardNo
0604218973495342
2720994062489249
341789811768403
342620005701588
3538317217289274
36398552430445
36468784920669
36546265094102
370986433966258
375871307825746
4485654034309174453
4508650093963164
4508811277155833
4556219373590294
4762842815339846
5020214085615419
5020717174306831
5176404203304135
5206903432150247
5419294263328638
5454472700716534
5497338640988625
5567695541477742
6011468308218604
6377502137509086
6380704194000620

Date of Transaction: Wednesday, May 09, 2018
Amount: 1000
Country: Madagascar
Transaction Type: Deposit
Merchant Category: Veterinary Services
Credit Card No: 5020214085615419
PAN No: AABCB1518L
Method: ● K-Means ○ SVM

Figure 3. Add new transaction



CreditCardNo
0604218973495342
2720994062489249
341789811768403
342620005701588
3538317217289274
36398552430445
36468784920669
36546265094102
370986433966258
375871307825746
4485654034309174453
4508650093963164
4508811277155833
4556219373590294
4762842815339846
5020214085615419
5020717174306831
5176404203304135
5206903432150247
5419294263328638
5454472700716534
5497338640988625
5567695541477742
6011468308218604
6377502137509086

```
[0]
1.0  4762842815339850.0  1.0
[1]
2.0  4556219373590290.0  2.0
[2]
3.0  4485654034309170000.0
[3]
4.0  2720994062489250.0  2.0
[4]
5.0  5206903432150250.0  1.0
[5]
6.0  5419294263328640.0  2.0
[6]
7.0  375871307825746.0  1.0  1
[7]
8.0  342620005701588.0  1.0  1
[8]
9.0  341789811768403.0  1.0  1
[9]
10.0  5454472700716530.0  2.0
[10]
11.0  5497338640988630.0  1.0
[11]
12.0  5454472700716530.0  2.0
```

Cluster Size: 4

Max Count: 100

Clustering in internal format:

```
332333000333320333330332303000111000030301013110З
```

Low Cluster

```
7.0 375871307825746.0 1.0 15000
8.0 342620005701588.0 1.0 18000
9.0 341789811768403.0 1.0 19000
15.0 341789811768403.0 1.0 1700
21.0 375871307825746.0 1.0 1600
26.0 604218973495342.0 1.0 1200
28.0 36398552430445.0 1.0 19000
29.0 36546265094102.0 1.0 14000
```

High Cluster

```
31.0 6399986262381010.0 1.0 140
32.0 6399693850187530.0 1.0 190
33.0 6377502137509090.0 1.0 200
42.0 6011468308218600.0 1.0 140
44.0 6011468308218600.0 1.0 180
46.0 6011468308218600.0 2.0 180
47.0 6011468308218600.0 1.0 120
52.0 6011468308218600.0 1.0 230
```

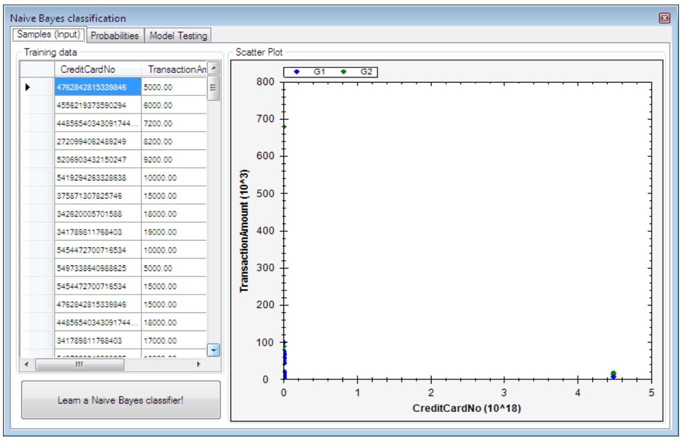
Risky Cluster

```
3.0 4485654034309170000.0 1.0 7200
14.0 4485654034309170000.0 1.0 180
24.0 4485654034309170000.0 1.0 130
```

High Risky Cluster

```
1.0 4762842815339850.0 1.0 5000
2.0 4556219373590290.0 2.0 6000
4.0 2720994062489250.0 2.0 8200
5.0 5206903432150250.0 1.0 9200
6.0 5419294263328640.0 2.0 1000
10.0 5454472700716530.0 2.0 100
11.0 5497338640988630.0 1.0 500
12.0 5454472700716530.0 2.0 150
```

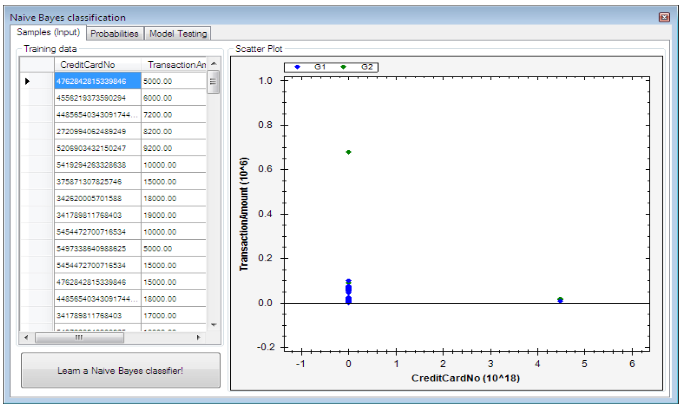
Figure 4. clustering result by K-Means



Training data

	CreditCardNo	TransactionAn
▶	4762842815339846	5000.00
	4556219373590294	6000.00
	44856540343091744...	7200.00
	2720994062489249	8200.00
	5206903432150247	9200.00
	5419294263328638	10000.00
	375871307825746	15000.00
	342620005701588	18000.00
	341789811768403	19000.00
	5454472700716534	10000.00
	5497338640988625	5000.00
	5454472700716534	15000.00
	4762842815339846	15000.00
	44856540343091744...	18000.00
	341789811768403	17000.00

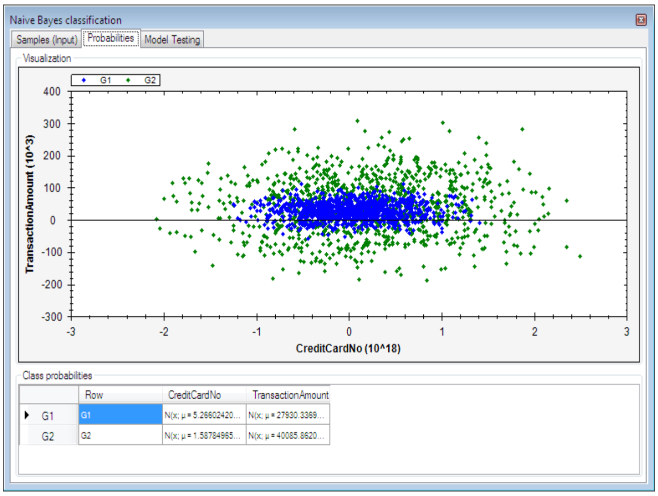
Figure 5. Apply naive Bayes Classifier



Training data

	CreditCardNo	TransactionAn
▶	4762842815339846	5000.00
	4556219373590294	6000.00
	44856540343091744...	7200.00
	2720994062489249	8200.00
	5206903432150247	9200.00
	5419294263328638	10000.00
	375871307825746	15000.00
	342620005701588	18000.00
	341789811768403	19000.00
	5454472700716534	10000.00
	5497338640988625	5000.00
	5454472700716534	15000.00
	4762842815339846	15000.00
	44856540343091744...	18000.00
	341789811768403	17000.00

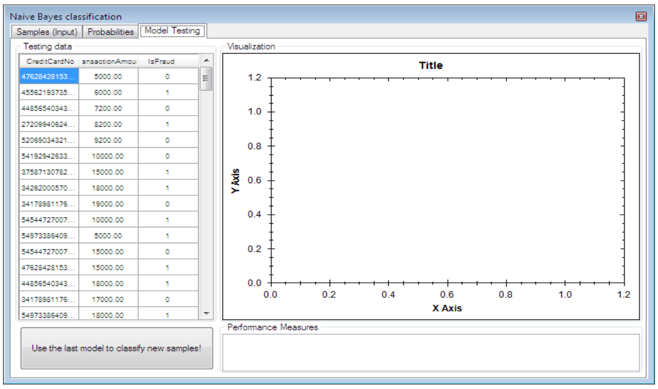
Figure 6. Load all Transaction Data



Class probabilities

	Row	CreditCardNo	TransactionAmount
▶ G1	G1	N(x; μ = 5.26602420...	N(x; μ = 27930.3369...
G2	G2	N(x; μ = 1.58784965...	N(x; μ = 40085.8620...

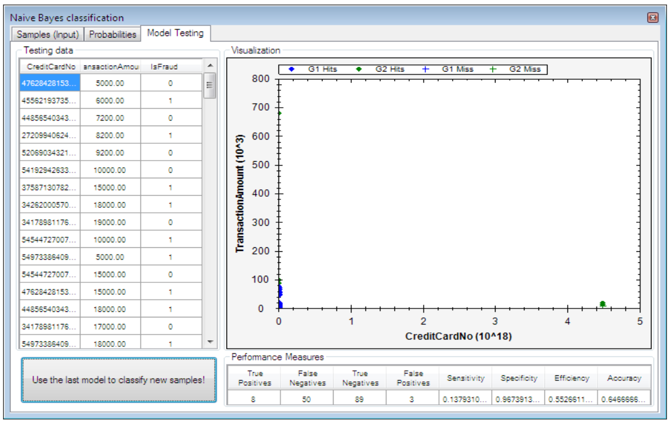
Figure 7. Probalities of Naïve bayes



Testing data

CreditCardNo	ansactionAmou	IsFraud
47628428153...	5000.00	0
45562193735...	6000.00	1
44856540343...	7200.00	0
27209940624...	8200.00	1
52069034321...	9200.00	0
54192942633...	10000.00	0
37587130782...	15000.00	1
34262000570...	18000.00	1
34178981176...	19000.00	0
54544727007...	10000.00	1
54973386409...	5000.00	1
54544727007...	15000.00	0
47628428153...	15000.00	1
44856540343...	18000.00	1
34178981176...	17000.00	0
54973386409...	18000.00	1

Figure 8. Model testing



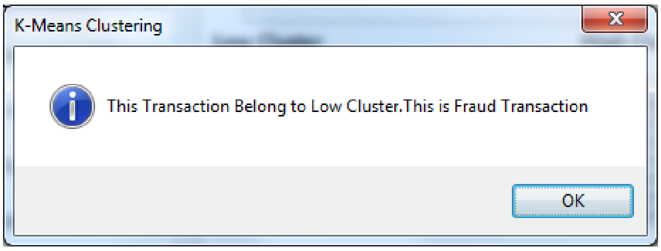
Testing data

CreditCardNo	ansactionAmou	IsFraud
47628428153...	5000.00	0
45562193735...	6000.00	1
44856540343...	7200.00	0
27209940624...	8200.00	1
52069034321...	9200.00	0
54192942633...	10000.00	0
37587130782...	15000.00	1
34262000570...	18000.00	1
34178981176...	19000.00	0
54544727007...	10000.00	1
54973386409...	5000.00	1
54544727007...	15000.00	0
47628428153...	15000.00	1
44856540343...	18000.00	1
34178981176...	17000.00	0
54973386409...	18000.00	1

Perfomance Measures

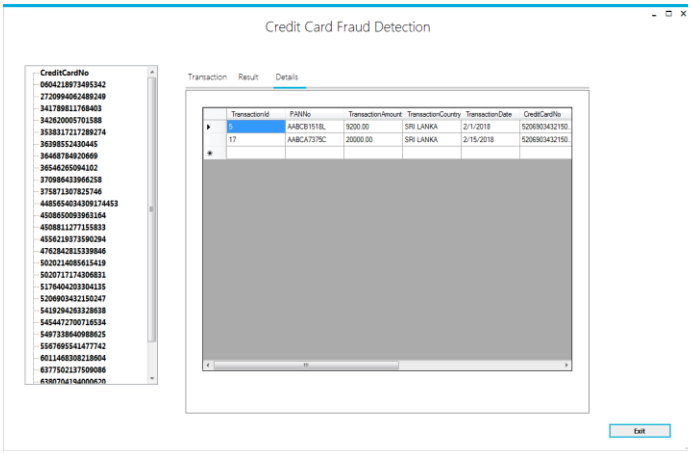
True Positives	False Negatives	True Negatives	False Positives	Sensitivity	Specificity	Efficiency	Accuracy
8	50	89	3	0.1379310...	0.9673913...	0.5526611...	0.6466666...

Figure 9. Naive Classification Predictive Result



This Transaction Belong to Low Cluster.This is Fraud Transaction

Figure 10. Transaction result

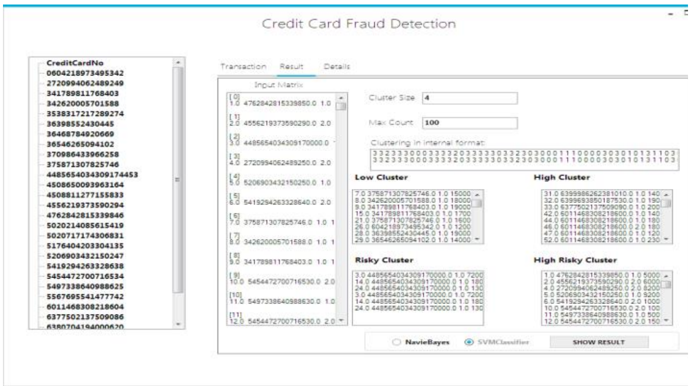


Credit Card Fraud Detection

CreditCardNo
0604218973495342
2720994062489249
341789811768403
342620005701588
353831721728927‌4
36398552430445
36468784920669
36546265094102
370986433966258
375871307825746
4485654034309174453
4508650093963164
4508811277155833
4556219373590294
4762842815339846
5020214085615419
5020717174306831
5176404203304135
5206903432150247
5419294263328638
5454472700716534
5497338640988625
5567695541477742
6011468308218604
6377502137509086
6380704194000620

TransactionId	PANNo	TransactionAmount	TransactionCountry	TransactionDate	CreditCardNo
5	AABCB1518L	9200.00	SRI LANKA	2/1/2018	5206903432150...
17	AABCA7375C	20000.00	SRI LANKA	2/15/2018	5206903432150...

Figure 11. Transaction and credit card details



Credit Card Fraud Detection

CreditCardNo
0604218973495342
2720994062489249
341789811768403
342620005701588
353831721728927‌4
36398552430445
36468784920669
36546265094102
370986433966258
375871307825746
4485654034309174453
4508650093963164
4508811277155833
4556219373590294
4762842815339846
5020214085615419
5020717174306831
5176404203304135
5206903432150247
5419294263328638
5454472700716534
5497338640988625
5567695541477742
6011468308218604
6377502137509086
6380704194000620

Input Matrix

```
[0]
1.0  4762842815339850.0  1.0
[1]
2.0  4556219373590290.0  2.0
[2]
3.0  4485654034309170000.0
[3]
4.0  2720994062489250.0  2.0
[4]
5.0  5206903432150250.0  1.0
[5]
6.0  5419294263328640.0  2.0
[6]
7.0  375871307825746.0  1.0  1
[7]
8.0  342620005701588.0  1.0  1
[8]
9.0  341789811768403.0  1.0  1
[9]
10.0  5454472700716530.0  2.0
[10]
11.0  5497338640988630.0  1.0
[11]
12.0  5454472700716530.0  2.0
```

Cluster Size: 4

Max Count: 100

Clustering in internal format:

```
3 3 2 3 3 3 0 0 0 3 3 3 3 2 0 3 3 3 3 3 0 3 3 2 3 0 3 0 0 0 1 1 1 0 0 0 0 3 0 3 0 1 0 1 3 1 1 0 3
3 3 2 3 3 3 0 0 0 3 3 3 3 2 0 3 3 3 3 3 0 3 3 2 3 0 3 0 0 0 1 1 1 0 0 0 0 3 0 3 0 1 0 1 3 1 1 0 3
```

Low Cluster

```
7.0 375871307825746.0 1.0 15000
8.0 342620005701588.0 1.0 18000
9.0 341789811768403.0 1.0 19000
15.0 341789811768403.0 1.0 1700
21.0 375871307825746.0 1.0 1600
26.0 604218973495342.0 1.0 1200
28.0 36398552430445.0 1.0 19000
29.0 36546265094102.0 1.0 14000
```

High Cluster

```
31.0 6399986262381010.0 1.0 140
32.0 6399693850187530.0 1.0 190
33.0 6377502137509090.0 1.0 200
42.0 6011468308218600.0 1.0 140
44.0 6011468308218600.0 1.0 180
46.0 6011468308218600.0 2.0 180
47.0 6011468308218600.0 1.0 120
52.0 6011468308218600.0 1.0 230
```

Risky Cluster

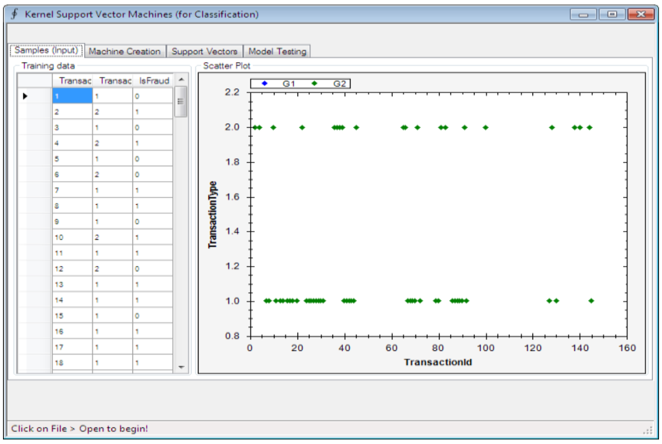
```
3.0 4485654034309170000.0 1.0 7200
14.0 4485654034309170000.0 1.0 180
24.0 4485654034309170000.0 1.0 130
3.0 4485654034309170000.0 1.0 7200
14.0 4485654034309170000.0 1.0 180
24.0 4485654034309170000.0 1.0 130
```

High Risky Cluster

```
1.0 4762842815339850.0 1.0 5000
2.0 4556219373590290.0 2.0 6000
4.0 2720994062489250.0 2.0 8200
5.0 5206903432150250.0 1.0 9200
6.0 5419294263328640.0 2.0 1000
10.0 5454472700716530.0 2.0 100
11.0 5497338640988630.0 1.0 500
12.0 5454472700716530.0 2.0 150
```

○ NavieBayes ● SVMClassifier

Figure 12. Load another Transaction



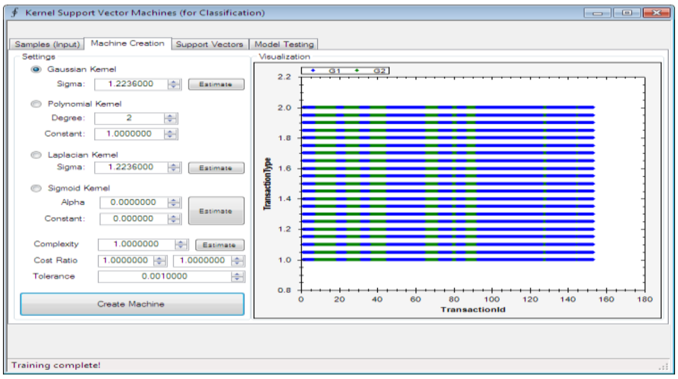
Kernel Support Vector Machines (for Classification)

Training data

Transac	Transac	IsFraud
1	1	0
2	2	1
3	1	0
4	2	1
5	1	0
6	2	0
7	1	1
8	1	1
9	1	0
10	2	1
11	1	1
12	2	0
13	1	1
14	1	1
15	1	0
16	1	1
17	1	1
18	1	1

Click on File > Open to begin!

Figure 13. Support Vector Machine Load Training Set



Kernel Support Vector Machines (for Classification)

Settings

● Gaussian Kernel
Sigma: 1.2236000

○ Polynomial Kernel
Degree: 2
Constant: 1.0000000

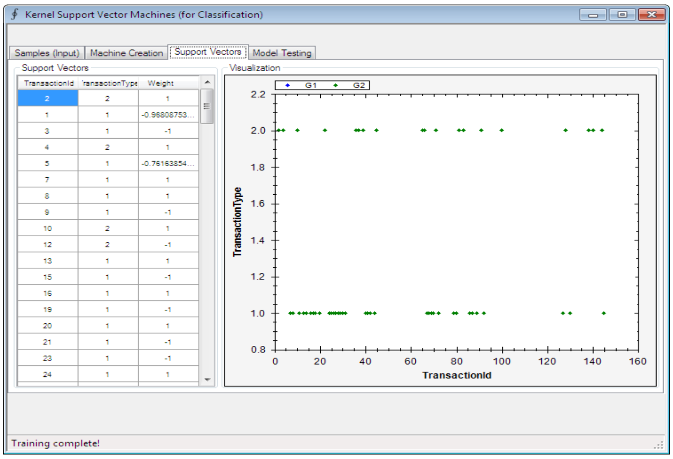
○ Laplacian Kernel
Sigma: 1.2236000

○ Sigmoid Kernel
Alpha 0.0000000
Constant: 0.000000

Complexity 1.0000000
Cost Ratio 1.0000000 1.0000000
Tolerance 0.0010000

Training complete!

Figure 14. Generate Training Model



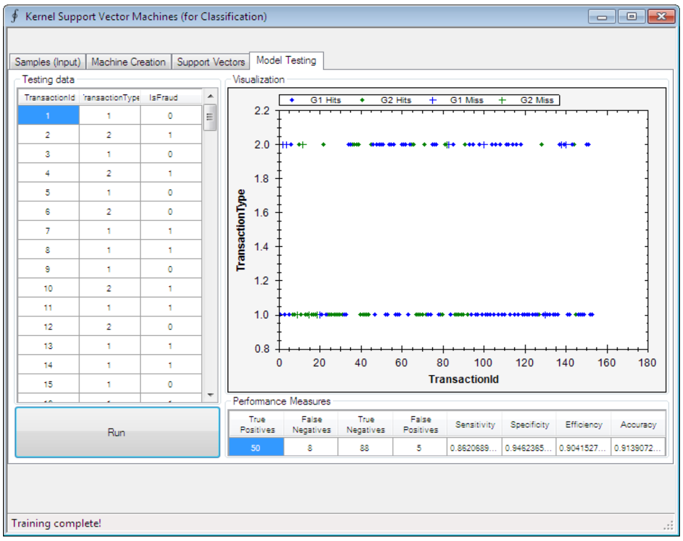
Kernel Support Vector Machines (for Classification)

Support Vectors

TransactionId	'ransactionType	Weight
2	2	1
1	1	-0.96808753...
3	1	-1
4	2	1
5	1	-0.76163854...
7	1	1
8	1	1
9	1	-1
10	2	1
12	2	-1
13	1	1
15	1	-1
16	1	1
19	1	-1
20	1	1
21	1	-1
23	1	-1
24	1	1

Training complete!

Figure 15. Training Weight Calculations



Kernel Support Vector Machines (for Classification)

Testing data

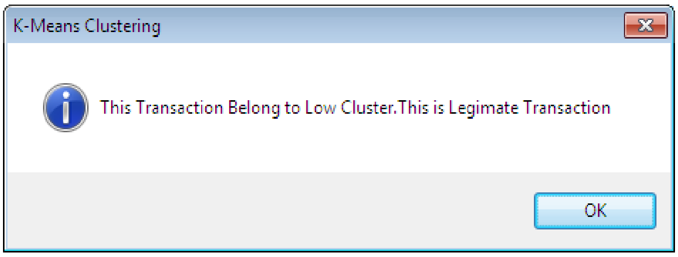
TransactionId	'ransactionType	IsFraud
1	1	0
2	2	1
3	1	0
4	2	1
5	1	0
6	2	0
7	1	1
8	1	1
9	1	0
10	2	1
11	1	1
12	2	0
13	1	1
14	1	1
15	1	0

Performance Measures

True Positives	False Negatives	True Negatives	False Positives	Sensitivity	Specificity	Efficiency	Accuracy
50	8	88	5	0.8620689...	0.9462365...	0.9041527...	0.9139072...

Training complete!

Figure 16. Predicting Using Testing Data



K-Means Clustering

This Transaction Belong to Low Cluster.This is Legimate Transaction

Figure 17. Transaction Result

Classified Data	NAVIE BAYES	SVM
TRUE POSITIVE	8	50
FALSE NAGATIVE	50	5
TRUE NAGETIVE	89	88
FALSE POSITIVE	3	5

Figure 18. COMPARITIVE RESULT based on Classified Data

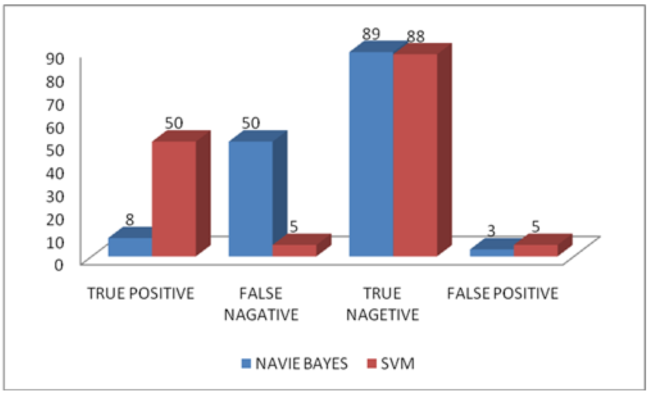


Figure 19. COMPARITIVE RESULT based on Classified Data in Graph

	NAVIE BAYES	SVM
Sensitivity	13.79	86.2
Specificity	96.73	94.41
Efficiency	55.26	90.41
Accuracy	64.66	91.31

Figure 20. Comparative result based on Accuracy

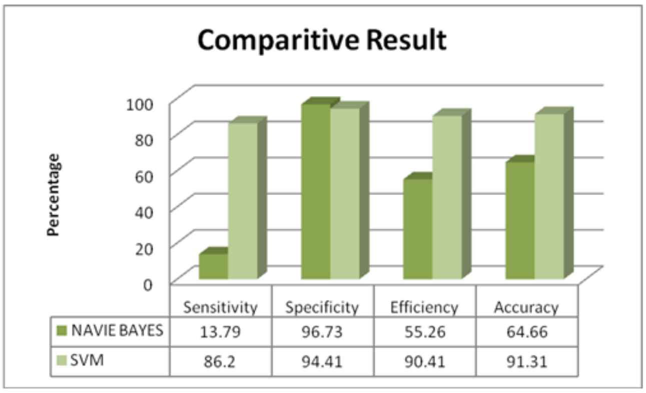


Figure 21. Comparative result based on Accuracy in Graph

V. CONCLUSION

Accomplishing the fraud detection accurately is a difficult task. However some errors were present in detecting the fraud. It is not possible that chances for fraud are always there. The transaction is may be at high risk but we cannot determine that it fraud or legitimate transaction. Fraudulent activities can be identified correctly in most of time. But some non-fraudulent activities can be detected as frauds. In this paper, combining Clustering and Classification approach is utilized for credit card fraud detection. Information is produced arbitrarily for credit card and after that K- means clustering calculation is utilized for recognizing the transaction whether it is misrepresentation or real. Clusters are framed to recognize detect fraud in credit card transaction exchange which are low, high, dangerous and high unsafe After applying Clustering introduced naïve bayes, SVM on highly skewed credit card fraud data. The two techniques are applied on the raw and preprocessed data. The results shows of optimal accuracy for naïve bayes, SVM neighbor classifiers are 64.66%, 91.31% respectively. The comparative results show that SVM performs better than naïve bayes techniques.

VI. REFERENCES

[1]. Maes, S., Tuyls, K., Vanschoenwinkel, B. and Manderick, B., (2002). Credit card fraud

detection using Bayesian and neural networks. Proceeding International NAISO Congress on Neuro Fuzzy Technologies.

[2]. Ogwueleka, F. N., (2011). Data Mining Application in Credit Card Fraud Detection System, Journal of Engineering Science and Technology, Vol. 6, No. 3, pp. 311 – 322

[3]. RamaKalyani, K. and UmaDevi, D., (2012). Fraud Detection of Credit Card Payment System by Genetic Algorithm, International Journal of Scientific & Engineering Research, Vol. 3, Issue 7, pp. 1 – 6, ISSN 2229-5518

[4]. Meshram, P. L., and Bhanarkar, P., (2012). Credit and ATM Card Fraud Detection Using Genetic Approach, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 10, pp. 1 – 5, ISSN: 2278-0181

[5]. Singh, G., Gupta, R., Rastogi, A., Chandel, M. D. S., and Riyaz, A., (2012). A Machine Learning Approach for Detection of Fraud based on SVM, International Journal of Scientific Engineering and Technology, Volume No.1, Issue No.3, pp. 194-198, ISSN : 2277-1581

[6]. Seeja, K. R., and Zareapoor, M., (2014). FraudMiner: A Novel Credit Card Fraud Detection Model Based on Frequent Itemset Mining, The Scientific World Journal, Hindawi Publishing Corporation, Volume 2014, Article ID 252797, pp. 1 – 10, http://dx.doi.org/10.1155/2014/252797

[7]. Patil, S., Somavanshi, H., Gaikwad, J., Deshmane, A., and Badgujar, R., (2015). Credit Card Fraud Detection Using Decision Tree Induction Algorithm, International Journal of Computer Science and Mobile Computing (IJCSMC), Vol.4, Issue 4, pp. 92-95, ISSN: 2320-088X

[8]. Duman, E., Buyukkaya, A., & Elikucuk, I. (2013). A novel and successful credit card fraud detection system implemented in a turkish bank. In Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on (pp. 162-171). IEEE.

[9]. Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2014). Improving credit card fraud detection with calibrated probabilities. In Proceedings of the 2014 SIAM International Conference on Data Mining (pp. 677-685). Society for Industrial and Applied Mathematics.

[10]. Ng, A. Y., and Jordan, M. I., (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Advances in neural information processing systems, 2, 841-848.

[11]. Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002). Credit card fraud detection using Bayesian and neural networks. In Proceedings of the 1st international naiso congress on neuro fuzzy technologies (pp. 261-270).

[12]. Shen, A., Tong, R., & Deng, Y. (2007). Application of classification models on credit card fraud detection. In Service Systems and Service Management, 2007 International Conference on (pp. 1-4). IEEE.

[13]. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.

[14]. Sahin, Y. 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.

[15]. Chaudhary, K. and Mallick, B., (2012). Credit Card Fraud: The study of its impact and detection techniques, International Journal of Computer Science and Network (IJCSN), Volume 1, Issue 4, pp. 31 – 35, ISSN: 2277-5420