Research paper on An improved credit card fraud detection using machine Learning models ¹Mr. Yogesh Kushwaha, ²Niraj Kumar, ³Rochit Shukla, ⁴Shivendra Srivastava

¹Mr. Yogesh Kushwaha, ²Niraj Kumar, ³Rochit Shukla, ⁴Shivendra Srivastava ¹Assistant Professor,Srm Institute Of Science And Technology ^{2,3,4}Students,Srm Institute Of Science And Technology

Abstract-The usage of credit cards for online and regular purchases is exponentially increasing and so is the fraud related with it. A large number of fraud transactions are made every day. Various modern techniques like Data Mining, Genetic Programming, etc. are used in detecting fraudulent transactions. This paper uses genetic algorithm which comprises of techniques for finding optimal solution for the problem and implicitly generating the result of the fraudulent transaction. The main aim is to detect the fraudulent transaction and to develop a method of generating test data. This algorithm is a heuristic approach used to solve high complexity computational problems. It is an optimization technique and evolutionary search based on the genetic and natural selection. The implementation of an efficient fraud detection system is imperative for all credit card issuing companies and their clients to minimize their losses.

The speedy participation in online primarily based transactional activities raises the fallacious cases everywhere and causes tremendous losses to the personal and financial business. [1] Although several criminal activities are occurring in commercial business, fraudulent e-card activities are among the foremost prevailing and disturbed regarding by online customers. Data processing techniques were used to check the patterns and characteristics of suspicious and nonsuspicious transactions supported normalized and anomalies knowledge. On the opposite hand, machine learning (ML) techniques were used to predict the suspicious and nonsuspicious transactions mechanically by victimization classifiers [2][5]. This paper discusses the supervised based mostly classification. When preprocessing the dataset using normalization and Principal element Analysis, all the classifiers achieved over 95.0% accuracy compared to results reached before preprocessing the dataset.

Keywords—*ML*; *Classification*; *Data processing*; *supervised*; *learning*.

I. INTRODUCTION

As businesses still move into the online community which currency is transacted dynamically in cash-less banking finance, adequate anomalies detection stay an important factor for bank systems. Not for the reason to stop the explicit cost obtained with counterfeit activities although verify that automated and manual reviews don't adversely wedge legitimate customers [3]. In deposit or withdraw trade, illegal transactions on card happens once someone abducts information from the card to undertake to purchases while no permission given from the holder and conjointly the detection of these dishonourable transactions has become a significant activity for payment processors.

A typical fraud detection systems encompass associate academic degree automatic tool and a manual technique. The automatic tool depends on fraud detection rules. It analyses all the new incoming transactions and assigns a fallacious score.

Fraud investigators produce the manual technique [6]. They concentrate on transactions with a high fallacious score and turn back binary feedback (fraud or legal) on all analysed activity. The fraud detection systems are supported professionally driven rules, knowledge-driven rules or a combination of every style of rules [4].

The created rules try to verify specific things of fraud discovered by the fraud investigators. A state of affairs of fraud is "a cardholder can avoid dealing throughout a given country and, among the 2 next weeks, he can another dealing for a given amount in another given country. If this example is detected among transactions, then the anomaly detection system will manufacture an academic degree alert. Machine learning algorithms rules. They learn the fallacious patterns and check out to find them during a data-stream of new incoming transactions. The usually used machine learning algorithms embody supply regression, SVM Fraud detection may be a problematic machine learning for many reasons.

Fraud can be defined as wrongful or criminal deception intended to result in financial or personal gain [1], or to damage another individual without necessarily leading to direct legal consequences. The two main mechanisms to avoid frauds and losses due to fraudulent activities are fraud prevention and fraud detection systems. Fraud prevention is the proactive mechanism with the goal of disabling the occurrence of fraud.

Fraud detection systems come into play when the fraudsters surpass the fraud prevention systems and start a fraudulent transaction. Nobody can understand whether a fraudulent transaction has passed the prevention mechanisms.

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

Accordingly, the goal of the fraud detection systems is to check every transaction for the possibility of being fraudulent regardless of the prevention mechanisms, and to identify fraudulent ones as quickly as possible after the fraudster has begun to perpetrate a fraudulent transaction. A review of the fraud detection systems can be found in [2-5].

With the developments in the information technology and improvements in the communication channels, fraud is spreading all over the world with results of huge financial losses. Though fraud can be perpetrated through many types of media, including mail, wire, phone and the Internet, online media such as Internet are the most popular ones. Because of the international availability of the web and ease with which users can hide their location and identity over Internet transactions, there is a rapid growth of committing fraudulent actions over this medium. Furthermore, with the improvements in the bandwidth of internetworking channels, fraudsters have the chance to form fraud networks among themselves through information change and collaboration all over the world. As a result, frauds committed over internet such as online credit card frauds become the most popular ones because of their nature.

Credit card frauds can be made in many ways such as simple theft, application fraud, counterfeit cards, never received issue (NRI) and online fraud (where the card holder is not present). In online fraud, the transaction is made remotely and only the card's details are needed. A manual signature, a

PIN or a card imprint are not required at the time of purchase. Though prevention mechanisms like CHIP&PIN decrease the fraudulent activities through simple theft, counterfeit cards and NRI; online frauds (internet and mail order frauds) are still increasing in both amount and number of transactions. There has been a growing amount of financial losses due to credit card frauds as the usage of the credit cards become more and more common. Many papers reported huge amounts of losses in different countries [2, 6-7]. According to Visa reports about European countries, about 50% of the whole credit card fraud losses in 2008 are due to online frauds.

II. STRUCTURE OF THE CREDIT CARD DATA

The credit card data used in this study are taken from a national bank's credit card data warehouses with the required permissions. The past data in the credit card data warehouses are used to form a data mart representing the card usage profiles of the customers. Though some of the customers may have more than one credit card, each card is taken as a unique profile because customers with more than one card generally use each card for a different purpose. Every card profile consists of variables each of which discloses a behavioral characteristic of the customers with respect to geographical locations, days of the month, hours of the day or merchant category codes (MCC) which show the type of the merchant

ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)

where the transaction takes place. Later on, these variables are used to build a model to be used in the fraud detection systems to distinguish fraudulent activities which show significant deviations from the card usage profile stored in the data-mart.

The number of transactions for each card differs from one to other; however, each transaction record is of the same fixed length and includes the same fields. Hand and Blunt gave a detailed description of the characteristics of credit card data [11]. These fields range from the date and hour of the transaction to the amount, transaction type, MCC code, address of the merchant where the transaction is done and etc. The date and hour of the transaction record shows when the transaction is made. Transaction type shows whether this transaction is a purchase or a cash-advance transaction. MCC code shows the type of the merchant store where the transaction takes place.

These are fixed codes given by the members of the VISA International Service Association. However; however, many of these codes form natural groups. So, instead of working with hundreds of codes, we grouped them into 25 groups according to their nature and the risk of availability to commit a fraud. The goods or services bought from merchant stores in some MCC codes can be easily converted to cash. As a result, transactions belonging to these MCC codes are more open to fraud and more risky from the transactions belonging to others.

The grouping of the MCC codes are done according to both the number of the fraudulent transactions made belonging to each MCC code and the interviews done with the personnel of the data supplier bank with domain expertise about the subject.

The distribution of our credit card data with respect to being normal or fraudulent is highly imbalanced with a ratio of about 20000 normal transaction records to one fraudulent transaction record. So, to enable the models to learn both types of profiles, some under sampling or oversampling techniques should be used. Instead of oversampling the fraudulent records by making multiple copies or etc., we use stratified sampling to under sample the legitimate records to a meaningful number.

Firstly, we identify the variables which show the most different distributions w.r.t. being fraudulent or normal. Then, we use these variables as the key variables in stratified sampling so that the characteristics of their distributions w.r.t. being fraudulent or not remains same.

- A. Advantages-
- *1)* The detection of the fraud is found much faster than the existing system.
- 2) In case of the existing system even the original card holder is also checked for fraud detection. But in this system no need to check the original user as we maintain a log.

3) We can find the most accurate detection using this technique.

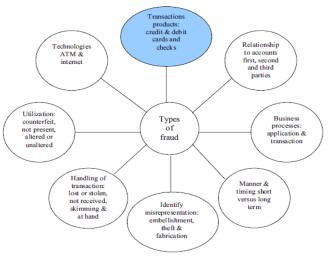


Fig.1: Types of fraud

- 4) This reduces the tedious work of an employee in the *Application*
- B. Application-
- 1) Provide easy and well security to Online Shopping.
- 2) Detect Frauds and trace the location from where the transaction has been made.
- C. Problems With Credit Card Fraud Detection-

There are lots of issues that makethis procedure tough to implement and one of the biggest problems associated with fraud detection is the lack of both the literature providing experimental results and of real world data for academic researchers to perform experiments on. The reasonbehind this is the sensitive financial data associated with the fraud that has to be kept confidential for the purpose of customer's privacy. Now, here we enumerate different properties a fraud detection system should have in order to generate proper results:

- 1) The system should be able to handle skewed distributions, since only a very small percentage of all credit card transactions is fraudulent.
- 2) There should be a proper means to handle the noise. Noise is the errors that is present in the data, for example, incorrect dates. This noise in actual data limits the accuracy of generalization that can be achieved, irrespective of how extensive the training set is.
- 3) Another problem related to this field is overlapping data. Many transactions may resemble fraudulent transactions when actually they are genuine transactions. The opposite also happens, when a fraudulent transactions appears to be genuine.
- 4) The systems should be able to adapt themselves to new kinds of fraud. Since after a while, successful fraud techniques decreases in efficiency due to the fact that they become well known because an efficient

fraudster always find a new and inventive ways of performing his job.

- 5) There is a need for good metrics to evaluate the classifier system. For example, the overall accuracy is not suited for evaluation on a skewed distribution, since even with a very high accuracy; almost all fraudulent transactions can be misclassified.
- 6) The system should take care of the amount of money that is being lost due to fraud and the amount of money that will be required to detect that fraud. For example, no profit is made by stopping a fraudulent transaction that is way lesser than the amount of money that will be required to detect it.
- 7) These points directus to the most important necessity of the fraud detection system, which is, a decision layer. The decision layer decides what action to take when fraudulent behavior is observed taking into account factors like, the frequency and amount of the transaction.
- D. Credit Card Fraud Detection Methods-

The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud. This model is then used to identify whether a new transaction is fraudulent or not. Our aim here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications.

Credit card fraud detection methods On doing the literature survey of various methods for fraud detection we come to the conclusion that to detect credit card fraud there are multiple approaches like:

- 1) Logistic regression
- 2) AdaBoost
- 3) Gradient Boosting
- 4) Bagging
- 5) Random forest
- 6) Neural network
- E. Logistic Regression-

Logistic Regression Logistic Regression is a supervised classification method that returnsthe probability of binary dependent variable that is predicted from the independent variableof dataset that is logistic regression predict the probability of an outcome which has two values either zero or one, yes or no and false or true. Logistic regression has similarities to linear regressionbut as in linear regression a straight line is obtained, logistic regression shows curve. The use of one or several predictorsor independent variable is on what prediction is based, logistic regression produces logistic curves which plots the values between zero and one.

III. LITERATURE REVIEW

Table 1. A summary of studies investigating different statistical techniques in credit card fraud

INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)

Study	Country	Method	Details
Aleskerov et al. (1997)	Germany	Neural networks	Card-watch
Bently et al. (2000)	UK	Genetic programming	Logic rules and scoring process
Bolton & Hand (2002)	UK	Clustering techniques	Peer group analysis and break point analysis
Brause et al. (1999a)	Germany	Data mining techniques & neural networks	Data mining application combined probabilistic and neuro-adaptive approach
Chan et al. (1999)	USA	Algorithms	Suspect behavioral prediction
Dorronsoro etal. (1997)	Spain	Neural networks	Neural classifier
Ezawa & Norton (1996)	USA	Bayesian networks	Telecommunication industry
Fan et al. (2001)	USA	Decision tree	Inductive decision tree
Ghosh & Reilly (1994)	USA	Neural networks	FDS (fraud detection system)
Kim & Kim (2002)	Korea	Neural classifier	Improving detection efficiency and focusing on bias of training sample as in skewed distribution. To reduce "mis-detections".
Kokkinaki (1997)	Cyprus	Decision tree	Similarity tree based on decision tree logic
Leonard (1995)	Canada	Expert system	Rule-based Expert system for fraud detection (fraud modelling)
Maes et al. (2002)	USA	Bayesian networks & neural networks	Credit card industry, back-propagation of error signals
		ł	1

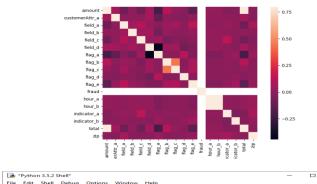
Kim & Kim (2002)	Korea	Neural classifier	Improving detection efficiency and focusing on bias of training sample as in skewed distribution. To reduce "mis-detections".
Kokkinaki (1997)	Cyprus	Decision tree	Similarity tree based on decision tree logic
Leonard (1995)	Canada	Expert system	Rule-based Expert system for fraud detection (fraud modelling)
Maes et al. (2002)	USA	Bayesian networks & neural networks	Credit card industry, back-propagation of error signals
Quah & Sriganesh (2007)	Singapore	Neural networks	Self-Organizing Map (SOM) through real-time fraud detection system
Wheeler & Aitken (2000)	UK	Combining algorithms	Diagnostic algorithms; diagnostic resolution strategies; probabilistic curve algorithm; best match algorithm; negative selection algorithms; density selection algorithms and approache
Zaslavsky & Strizkak (2006)	Ukraine	Neural networks	SOM, algorithm for detection of fraudulent operations in payment system

IV. RESULTS

This detection process constitutes of four steps. These steps are mentioned below-

- A. Input all the transactions record and standardize the data. Finally get the sample which includes the confidential information about the card holder in the data set with their consent.
- *B.* In this step the CCusage frequency count, CC location, CC overdraft, Current bank balance and average daily spending is computed.
- *C.* Generating critical values after finding out the limited number of generations for critical fraud detected, monitorable fraud detected, ordinary fraud detected, etc. using Genetic Algorithm.

D. Generate fraud transactions detected in the final step. It is done by applying detection mining on critical values obtained in the process of fraud detection.



×

<pre>File Exit Shel Debug Options Window Help 0:30 to 100 th 0:32: </pre>	
<pre>- Radom forest - I, Score () >0733356462462463 I, Score () >0733356463642463 AUC() 0.9202416341210767 FrecAll () 0.9202416341210767 FrecAll () 0.9202416341210767 FrecAll () 0.937316561694663 FrecAll () 0.9153566059427937 arGut acy () 0.78060600606 FrecAll () 0.7306608060606 FrecAll () 0.7306608060606 FrecAll () 0.731740041247212 Frets () 0.5352244727426136 FrecAll () 0.731740041247212 Frets () 0.5352244727426136 FrecAll () 0.731740041247212 Frets () 0.730670080807427937 FrecAll () 0.7306706080606 FrecAll () 0.7302560089377566 FrecAll () 0.730256008645703 FrecAll () 0.73000757857850 FrecAll () 0.73000757857850 FrecAll () 0.73000757857850 FrecAll () 0.73000757857850 FrecAll () 0.73000757856 FrecAll () 0.7200757850 FrecAll (</pre>	dit She <u>l</u> l <u>D</u> ebug <u>O</u> ptions <u>W</u> indow <u>H</u> elp
<pre>Fi_Boores 0.0073335564624629 Accuracy 0.05896400406000 recall 0.05896400406000 recall 0.05896400406000 recall 0.058564004963 Fils 0.0518564037427937 Accuracy 0.78564060000 recall 0.7117400419207312 File "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti 0.90% line 433 Warning (from varnings module): Fils "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti recall 0.702306079640503 Files 0.511240017627921 Warning (from varnings module): Fils "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti recall 0.7023060796465963 Accuracy 0.737686669666 Accuracy 0.737686669666 Accuracy 0.737686669666 Accuracy 0.737686669666 Accuracy 0.737686669666703 Fils "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Files/Fython35%lib/site-packages/sklearn/linear_model/logisti Fils "C(\Frogram Files/Files/Files/Files/Files/</pre>	to 100 in 0.22.
acGuracy1 0.98900000000001 AUC: 0.9301034120787 BHAC: 0.9301034120787 BHAC: 0.9301034120735003 FHEEA: 0.9101073350035 FHEEA: 0.900000040060000 AUC: 0.7500065746041 Feca: 0.930065746041 Feca: 0.930234473743010 Warning (from varings module): 0.92",line 433 FULLEWWarning) FULLEWWarning; Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Logistic regression - FL_Score: 0.501200494608900 AUC: 0.730260098977506 Fecal: 0.501200496460890 AUC: 0.730260098977506 Fecal: 0.501200495075201 Warning (from varings module): File "Civrogem Files/Python35/lib/site-packages/sklearn/linear_model/logisti 0.730260098977506 Fecal: 0.5012004660503 FHEE: 0.512240176278291 Warning (from varings module): File "Civrogem Files/Python35/lib/site-packages/sklearn/linear_model/logisti FulleWarning; Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting - Voting - Voting - Voting - Voting - Votin	dom forest -
ADC: 0.5252cl6341210767 FreeAll 0.625731665164666 - Neural network - Fi_Score: 0.515366057427537 Arguracy: 0.515366057427537 Arguracy: 0.515366057427537 FreeAll 0.517174004158275112 FreeAll 0.51717400415827512 FreeAll 0.51717400415827512 FreeAll 0.51717400415827512 FreeAll 0.51717400415827512 FreeAll 0.517240412 File *C:\Frogram Files\Fython35\Lib\site-packages\sklearn\linear_model\logisti -yy*, line 433 0.732650560566566 ACC: 0.7305260569567566 FreeAll 0.512540617657551 FreeAll 0.512540617657551 FreeAll 0.752060766266566 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.73052606666666 ACC: 0.730526066757550 FreeAll 0.612540617657551 FreeAll 0.612540617657551 FreeAll 0.612540617657551 FreeAll 0.612540617657551 FreeAll 0.612540617657555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.6125407555 FreeAll 0.612540755 FreeAll 0.6125407555 FreeAll 0.61254075555 FreeAll 0.6125555 FreeAll 0.61	ore: 0.9073335964624629
<pre>recall 0.00057427820 Facta 0.0005742782 Fm_Stat 0.00057427837 acturacy 0.00050000000 excutacy 0.0000000000 recall 0.7117400419207312 Facta 0.00052244727424316 Warning (from varnings module) 1 File "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FutureWarning; Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to sitence this varning. Files 0.501200406408050 accuracy 0.7623060796468950 accuracy 0.7623060796468950 Accuracy 0.7623060796468950 Accuracy 0.7623060796468950 Feetal 0.512240176279251 File "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti File "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files T(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files T(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files T(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\logisti Files Files Files "C(Frogram Files/Python35\lib\site-packages\sklearn\linear_model\sklearn\linear Files Files Files "C(Frogram Files/Python35\lib\site-packages\sklearn\linear Files Files Files Files Files Files Files Files Files Files F</pre>	acy: 0.9890606060606061
<pre>Fmetai 0.9151191073358055 - Neutai network - Neutai network - accuracy: 0.75000000000000 Accuracy: 0.750000574605401 Ferail 0.030124973140100 Neural 0.05000574605401 (from varings module): - Logitid from varings in a site will be changed to 'lbfgs' in 0.22. Specify a solv = continued in the site of the site</pre>	0,9282616341210767
<pre>- Neural network Neural Neura</pre>	1: 0.8637316561844863
<pre>Fi_Boores' 0.51550c6057427937 acGuracy 0.7505060050606 Frecall 0.7117400418207312 Frecal 0.55024472747426136 Warning (from varnings module)! File "C(!Frogram Files/Python35%lib\site-packages\sklearn\linear_model\logisti 6.py", line 433 multi-extra files/Python35%lib\site-packages\sklearn\linear_model\logisti File Science This varning. FileScience 10.501209044680802 acGuracy1 0.7578666580868686 acGuracy1 0.751866658686866 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 AcGuracy1 0.75786665868686 Field 0.512300176278231 Field 0.75786665878 Field 0.512300176278231 Field 0.75786665878 Field 0.512300176278231 Field 0.75786665878 Field 0.512300176278231 Field 0.512300000000000000000000000000000000000</pre>	: 0.9151191073358835
<pre>acouracy: 0.7800000000000 AUC: 0.7800005746051 FBC: 0.780005746051 FBC: 0.78005746051 FBC: 0.78005746051 FBC: 0.5302344727420136 Warning (rom warnings module)! FB: "C:\Frogram Files\Fython35\Lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FBC: 0.500500000000000000000000000000000000</pre>	ral network -
ADC: 0.7550006574660411 Frecal: 0.7517400415827113 FReta: 0.5512244727420136 Warning (From Warnings module): File "Ci\Frogram Files\Fython35\lib\site-packages\sklearn\linear_model\logisti FilesWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solve to silence this warning. FilesWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solve to silence this warning. FilesWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solve to silence this warning. FilesWarning: O.7576666668666666 AUC: 0.7507666686866666 AUC: 0.7507666686866666 AUC: 0.7507676767877876 AUC: 0.757666686866666 AUC: 0.75076767877876 AUC: 0.757666687678787 FilesWarning: Files\Fython35\lib\site-packages\sklearn\linear_model\logisti FilesWarning: Forwarning: solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting - this warning Voting - this warning	ore: 0.5193566097427937
<pre>recall 0.7117400418207312 FBets1 0.535244727424366 Warning (from warnings module): File *Ci(Frogram Files/Fython35%lib/site-packages/sklearn/linear_model/logisti File *Ci(Frogram Files/Python35%lib/site-packages/sklearn/linear_model/logisti c. 0.012009404680802 accuracy1 0.75746645456 accuracy1 0.75746645456 accuracy1 0.757466454563 File *Ci(Frogram Files/Python35%lib/site-packages/sklearn/linear_model/logisti File *Ci(Frogram F</pre>	acy: 0.786060606060606
<pre>Fmees. 0.5552244727426136 Warning (From warnings module)! File "Ci\Frogram Files\Fython35\lib\site-packages\sklearn\linear_model\logisti c.py", inter 433 FutureWarning) FutureWarning - Logistic regression - Formation (0.7305600696666666 ADC: 0.7305260069377566 Freesil: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607667039 Field (0.73086767693) Field (0.730867666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.73028607666666666 ADC: 0.730286076666666666 ADC: 0.730286076666666666 ADC: 0.730286076666666666 ADC: 0.730286076676666666666 ADC: 0.730286076676666666666 ADC: 0.73028676666666666 ADC: 0.73028676766666666666 ADC: 0.7302867676666666666 ADC: 0.73028676766666666666 ADC: 0.73028676766666666666 ADC: 0.7302867676666666666666666 ADC: 0.7302867676666666666666 ADC: 0.730286776666666666666666666 ADC: 0.73028676766666666666666666666666 ADC: 0.730286776666666666666666666666666666666666</pre>	0.750006574668411
<pre>Warning (from warnings module): File = Trive set in the s</pre>	1: 0.7117400419287212
<pre>File "C'\Program Files\Python35\lib\site-packages\sklearn\linear_model\logisti c,py", line 433 FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solve st to silence this warning Logistic represedon - acouracy: 0.75506566666666 AUC: 0.7502560766666666 AUC: 0.7502560766666666 AUC: 0.750256076666666 AUC: 0.750256076666666 AUC: 0.75025607667651 FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv ruprewarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voring - Voring = - To other warning: Solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voring Voring</pre>	1 0.5352244727426136
<pre>File "C'\Program Files\Python35\lib\site-packages\sklearn\linear_model\logisti c,py", line 433 FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solve st to silence this warning Logistic represedon - acouracy: 0.75506566666666 AUC: 0.7502560766666666 AUC: 0.7502560766666666 AUC: 0.750256076666666 AUC: 0.750256076666666 AUC: 0.75025607667651 FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv ruprewarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voring - Voring = - To other warning: Solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voring Voring</pre>	
<pre>c.pw; line 433 FutureWarning) FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv cognetic regressio - Fi_Score: 0.5012089494680802 accuracy: 0.73046864686866 recall: 0.7023060796648703 Freta: 0.512408176278201 Warning (from warnings module): File "Cf:Program Files\Fython35\lib\site-packages\klearn\linear_model\logisti FutureWarning) FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting-</pre>	
<pre>FutureWarning) FutureWarning Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Logistic regression - Fl_Score: 0.50120964680805 ACC: 0.7300260089377556 Fresal: 0.77023060796648703 Fresal: 0.77023060796648703 Fresal: 0.512340817637039 Fresal: 0.51234081763703 Fresal: 0.5123408176370 Fresal: 0.51234081763</pre>	e "C:\Program Files\Python35\lib\site-packages\sklearn\linear model\logisti
<pre>FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv to silence this warning. F. Jabrei 0.501200947 Saccuracy 0.501200947 Accuracy 0.501200947 Accuracy 0.501200947 August 0.501200947 Fetta 0.511240917627021 Werning (from warnings module): FutureWarning) FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solv = Voting this warning. FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solv = Voting this warning. FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solv = Voting this warning. FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solver = Voting this warning. FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solver = Voting this warning. FutureWarning: Solver will be changed to 'lbfgs' in 0.22. Specify a solver = Voting this warning. = Voting this warning this war</pre>	, line 433
er to slience this warning. - Logistic regressio Fl_Score: 0.5012006494609906 ACC: 0.7302360098977856 recall: 0.7023060786645703 FBete: 0.512400176278261 Warning (from warninge module): File "C(:NFrogram Files/Fython35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FutureWarning) FutureWarning: FutureWar	
- Logistic regression - - Logistic regression - Ti_Score: 0.50130064660908 ADC: 0.7302560098977586 Frecal: 0.57023060706645703 Frecal: 0.5122400176278291 Marning (from varinings module): - File "Ci\Frogum Files/Pyth0n35\lib\site-packages\sklearn\linear_model\logisti - FilewArning) FutureMarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting = this warning.	eWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv
<pre>Fl_Score: 0.5012009404680802 accuracy: 0.7576666666666666 ACC: 0.75766666767656 ACC: 0.7576666779656 PBets: 0.512400176579291 Warning (from warnings module): File "C(:VProgener Files/Python35\lib\site-packages\sklearn\linear_model\logisti c.pV", line 433 FutureWarning; Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting- this warning.</pre>	silence this warning.
acouracy1 0.757666656565666 ACC: 0.7302560095977566 Fectal:0.702306075675261 Warning (from warnings module): File "C:UProgram Files/Python35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 Futewarning) Futewarning: - Voting - Take Warning. - Voting - Voting	istic regression -
ADC: 0.7308260098377866 Tercall: 0.7023060796648703 Thete: 0.5123400746279291 Warning (from varnings module): File "Ci\Frogram Files\Python35\lib\site-packages\sklearn\linear_model\logisti FutureWarning: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Vocime - this warning.	ore: 0.5012089484688892
<pre>recal: 0.7023060756648703 FBEts: 0.512408176278251 Warning (from warnings module): File "CiVProgram Files/Fython35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FutureWarning) FutureWarning Fu</pre>	acy: 0.7576969696969696
<pre>FBeta: 0.5122408176279291 Warning (from warnings module): File "C:UProgram Files/Python35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 provide the state of the stat</pre>	0.7308260099377586
Variance (from warnings module): () it was a price of the state of th	1: 0.7023060796645703
File "C:\Program Files\Python35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FutureWarning) FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv er to silence this warning. - Voting -	: 0.5122408176279291
File "C:\Program Files\Python35\lib\site-packages\sklearn\linear_model\logisti c.py", line 433 FutureWarning) FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv er to silence this warning. - Voting -	
c.pw, line 433 FutureWarning) FutureWarning Default solver will be changed to 'lbfgs' in 0.22. Specify a solv - Voting -	
$^{-1}$ FutureWarning) FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv er to silence this warning. $-$ Voting $-$	
FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solv er to silence this warning. - Voting -	
er to silence this warning. - Voting -	
- Voting -	
	ore: 0.7154837333404711
accuracy: 0.9486969696969697	
AUC: 0.8520518068170088	
recall: 0.749475890985325 Activate V	
	: 0.7521808913843153 Go to Setting
	Go to Setting
Ln: 81 Co	Ln: 81 Col:

V. CONCLUSION

The importance of Machine Learning and Data Science cannot be overstated. If you are interested in studying past trends and training machines to learn with time how to define scenarios, identify and label events, or predict a value in the present or future, data science is of the essence. It is essential to study the underlying data and model it by selecting an appropriate algorithm to approach any such use case. The various control parameters of the algorithm need to be tweaked to fit the data set. As a result, the developed application improves and becomes more efficient in solving the problem.

Credit card fraud has been deeply rooted in the ecommerce industry. In this scenario more of the financial losses are associated with the e-commerce merchants. To save merchant from these losses we have proposed the Credit Card fraud risk assessment model. In order to improve fraud risk assessment we have used combination of two presented methods. In proposed model, genetic algorithm is applied on the clusters generated by Logistic regression algorithm. Genetic algorithm will optimize the output generated by Logistic regression.

The rule engine is used so that system is scalable in terms of rules. In future this model can be extended by adding various rules in rule engine to improve accuracy of the system.

INTERNATIONAL JOURNAL OF RESEARCH IN ELECTRONICS AND COMPUTER ENGINEERING

A UNIT OF I2OR

IJRECE VOL. 7 ISSUE 2 (APRIL- JUNE 2019)

ISSN: 2393-9028 (PRINT) | ISSN: 2348-2281 (ONLINE)

REFERENCES

- [1] Hobson, A.: The Oxford Dictionary of Difficult Words. The Oxford University Press, New York (2004)
- [2] Bolton, R. J., Hand, D. J.: Statistical fraud detection: A review. Statistical Science 28(3), 235--255 (2002)
- [3] Kou, Y., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P.: Survey of fraud detection techniques. In: Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan (2004)
- [4] Phua, C., Lee, V., Smith, K., Gayler, R.: A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review (2005)
- [5] Sahin, Y., Duman, E.: An overview of business domains where fraud can take place, and a survey of various fraud detection techniques. In: Proceedings of the 1st International Symposium on Computing in Science and Engineering, Aydin, Turkey (2010)
- [6] Leonard, K. J.: Detecting credit card fraud using expert systems. Computers and Industrial Engineering, 25, (1993)
- [7] Ghosh, S., Reilly, D. L.: Credit card fraud detection with a neural network. In: Proceedings of the 27th Hawaii International Conference on system Sciences, (1994)
- [8] Mena, J.: Investigate Data Mining for Security and Criminal Detection, Butterworth-Heinemann, Amsterdam (2003)
- [9] Aleskerov, E., Freisleben, B., Rao, B.: CARDWATCH: A neural network based data mining system for credit card fraud detection. In: Computational Intelligence for Financial Engineering, 220-226 (1997)
- [10] Chen, R., Chiu, M., Huang, Y., Chen, L.: Detecting credit card fraud by using questionnaire-responded transaction model based on SVMs. In: Proceedings of IDEAL2004 (2004)
- [11] Hand, D. J., Blunt, G.: Prospecting gems in credit card data. IMA Journal of Management Mathematics, 12 (2001)
- [12] Dahl, J.: Card Fraud. In: Credit Union Magazine (2006)
- [13] Schindeler, S.: Fighting Card Fraud in the USA. In: Credit Control, House of Words Ltd. (2006)
- [14] Brause, R., Langsdorf, T., Hepp, M.: Neural data mining for credit card fraud detection. In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (1999)
- [15] Chen, R.-C., Luo, S.-T., Liang, X., Lee, V. C. S.: Personalized approach based on SVM and ANN for detecting credit card fraud. In: Proceedings of the IEEE International Conference on Neural Networks and Brain, Beijing, China (2005)
- [16] Dorronsoro, J. R., Ginel, F., Sanchez, C., Cruz, C. S.: Neural fraud detection in credit card operations. IEEE Transactions on Neural Networks, 8 (1997)
- [17] Hanagandi, V., Dhar, A., Buescher, K.: Density-Based Clustering and Radial Basis Function Modeling to Generate Credit Card Fraud Scores. In: Proceedings of the IEEE/IAFE 1996 Conference (1996)
- [18] Juszczak, P., Adams, N. M., Hand, D. J., Whitrow, C., Weston, D. J.: Off-the-peg and bespoke classifiers for fraud detection. Computational Statistics & Data Analysis. 52(9) (2008)

- [19] Quah, J. T., Sriganesh, M.: Real-time credit card fraud detection using computational intelligence. Expert Systems with Applications. 35(4) (2008)
- [20] Shen, A., Tong, R., Deng, Y.: Application of classification models on credit card fraud detection. In: International Conference on Service Systems and Service Management, Chengdu, China (2007)
- [21] Syeda, M., Zhang, Y., Pan, Y.: Parallel granular neural networks for fast credit card fraud detection. In: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems (2002)