CREDIT CARD FRAUD DETECTION BASED ON DEEP NEURAL NETWORK APPROACH

V.Mallesi, M.Tech., Head of the Department, Department of CSE, csehod.bitsadoni@gmail.com
M.Keerthi, Btech, Department of CSE, matamkeerthi1998@gmail.com

ABSTRACT: Reality-based credit card fraud detection is the project's primary emphasis. There has been a dramatic increase in fraudulent acts as a result of the meteoric rise in the use of credit cards in recent years. The goal is to gain services or items without paying for them, or to steal money from an account without permission. All credit card issuing institutions must now have reliable fraud detection systems to cut down on losses. The fact that neither the card nor the cardholder need to be present during the transaction poses a significant obstacle to the success of the firm. Since the shop can't check the customer's identity, it's feasible that a fake card may be used. The suggested method employs Machine Learning techniques to increase the precision with which fraud is detected. The random forest method is used for classification, analyzing both the input dataset and the user's most recent input dataset. In the end, you should maximize the precision of the obtained information. Precision, sensitivity, and accuracy are the metrics used to assess the effectiveness of the methods. The graphical model is then seen after some of the given characteristics are processed to identify fraud. Accuracy, sensitivity, specificity, and precision are used to assess the methods' effectiveness.

Key words: Random Forest, Logistic regression, Support vector machine, Decision tree, and XGboost are some of the terms used to describe machine learning techniques.

1. INTRODUCTION

As a result of the migration The precise identification of fraud is a crucial aspect in protecting electronic monetary transactions, which have become more common due to the shift of company operations to the Internet and the expanding cashless economy. When a criminal makes purchases using a stolen credit card number, this is called credit card fraud. Credit card fraud costs businesses and consumers billions of dollars annually because of widespread usage and inadequate security measures. It is difficult to get an accurate estimate of the losses since credit card companies are often hesitant to reveal such statistics. Credit card fraud may result in significant financial losses, although some information about these losses is available to the public. Credit card fraud costs businesses and governments throughout the world billions of dollars annually. In 2017, credit card theft cost businesses and consumers across the world a total of $22.8 billion, and that figure is only anticipated to rise to $31 billion by 2020. Application fraud and use fraud make up the two broad types of credit card scams. In the context of credit cards, "application fraud" refers to fraudulent applications for new cards. If a criminal requests a new credit card transaction using stolen personal information and the issuer agrees, the criminal has committed identity theft. After a credit card is legitimately issued, fraudulent conduct during a purchase is considered behavior fraud. Identifying fraudulent activity on credit cards has been a major challenge for consumers and banks. Credit card fraud is also a major issue for academics, since the detection of even a small percentage of fraudulent transactions would safeguard substantial sums of money.

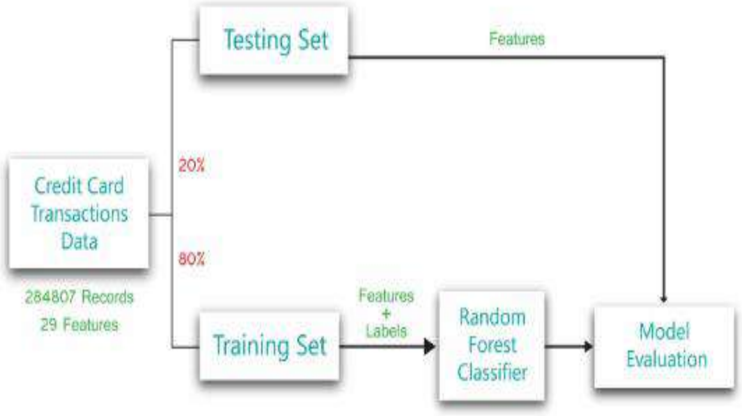


Fig.1: Example figure

2. LITERATURE REVIEW

2.1 HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture,

Each year, credit card theft costs issuers billions of dollars. Reducing fraud losses requires a well-developed fraud detection system, equipped with a state-of-the-art fraud detection model. Our key contribution is a method for detecting fraud that uses a deep learning architecture and an enhanced feature engineering approach based on the principle of behavioral homogeneity (HOBA). We undertake comparison research to evaluate the efficacy of the suggested framework using a real-world dataset from one of the top commercial banks in China. The experimental results demonstrate the efficiency and practicability of our suggested technique in detecting credit card fraud. In practice, our suggested technique is able to detect more fraudulent transactions than the benchmark methods while maintaining a false positive rate that is within acceptable limits. Our finding has important management implications since it suggests that the suggested approach may be used by credit card issuers to more effectively detect fraudulent transactions, therefore protecting consumers' interests and lowering fraud losses and regulatory expenses.

2.2 E-commerce credit card fraud detection via data mining

Losses from credit card theft cost internet businesses billions of dollars annually. Researchers have found more sophisticated approaches to identify fraud with the advancement of machine learning algorithms, but actual implementations are seldom documented. Here, we detail our experience creating and deploying a fraud detection system for a major online retailer. The research examines the feasibility of integrating human and automated categorization, details the whole procedure of development, and evaluates and contrasts several machine learning approaches. As a result, the study may aid academics and practitioners in developing data mining based systems for detecting fraud and other related issues. In addition to the automated system developed as part of this project, the fraud analysts have gained valuable insights on how to optimize the human revision process, leading to much better results.

2.3 In order to identify credit card fraud, deep neural networks have been trained to adapt to new domains (2.3).

Although credit card theft only affects a tiny fraction of all purchases, the resulting losses might be substantial. Because of this, automated Fraud Detection Systems (FDS) that can spot scams with pinpoint accuracy and adapt to the many methods used by fraudsters must be developed. Indeed, the kind of fraud committed may vary widely by nation, demographic group, and mode of payment (e.g., e-commerce vs store terminal). This comes down to the well-known issue of transfer learning, which is becoming more critical for transactional organizations as the cost of creating data-driven FDSs rises.

2.4 Using the local outlier factor and isolation forest to identify credit card fraud

Today's rapidly developing technologies may be used to both positive and negative uses. E-commerce and other types of online transactions, the majority of which are paid for using credit cards, have therefore increased as a result of the development of such technologies. Credit cards enable consumers to make immediate, interest-free purchases at brick-and-mortar stores and on the Internet without worrying about a payment plan. It allows customers to pay with their credit cards at any store, worldwide. Credit card fraud is on the rise as credit card use rises. The credit card processing mechanism is very susceptible to scams. Because of the enormous sums of money that are lost each year as a result of credit card fraud, criminals are always on the lookout for novel ways to conduct these acts of lawlessness. Banking and finance institutions have a significant challenge in identifying fraudulent online transactions. Therefore, it is of utmost importance for banks and financial institutions to have effective fraud detection systems in place to minimize losses brought on by credit card fraud transactions. Many academics have come up with different methods to identify and minimize these scams. In this research, we present a comparison of the Python-based Local Outlier Factor and Isolation Factor algorithms and provide extensive experimental findings contrasting the two. Using Local Outlier Factor, we were able to get an accuracy of 97%, whereas Isolation Forest only managed to get us to 76%.

2.5 Suspicion score for identity theft in real-time credit applications according to community analysis

The research proposes a speedy method, called communal analysis suspicion scoring (CASS), for assigning numerical suspicion scores to credit applications in real time based on implicit relationships between them across both time and location. CASS incorporates temporal and geographic weights and smoothed k-wise scoring of many connected application-pairs in addition to pair-wise communal scoring of identifying properties for applications. Several hundred thousand genuine credit applications were mined to show that CASS effectively lowers false alert rates while maintaining acceptable hit rates. Due to its scalability, CASS is able to quickly identify identity theft warning signs in this massive data set. Connections between software have also revealed surprising new perspectives.

Commercial Banks' Use of Support Vector Machines for Analyzing Credit Risk:

An index system is developed based on the current state of credit risk assessment in commercial banks. Both monetary and non-monetary indices are included in the index system. The evaluation in this study makes use of the SVM (support vector machine) technique. Training sets are chosen using the strategy based on rising proportions. A larger sample size yields more accurate percentage. Experimental findings demonstrate the model's high accurate classification accuracy, and a real-world instance is provided to validate the method's efficacy.

3. METHODOLOGY

- Every year, credit card theft costs businesses and individuals billions of dollars. As ancient as civilization itself, fraud exists in an infinite number of guises. According to the 2017 PwC Global Economic Crime Survey, over half of all businesses were victims of economic crime. As a result, there is undeniably a need to find a solution to the issue of credit card fraud detection. In addition, the proliferation of advanced technology has expanded the opportunities for fraud. Credit card fraud has been steadily rising over the last several years, reflecting the widespread usage of credit cards in today's society. Hugh Fraudulent financial losses impact not just businesses and financial institutions, but also consumers who use the credits. Merchants may also suffer intangible damages, such as damage to their reputation and brand image, as a result of fraud. If a customer experiences fraud when using his credit card with one provider, he may decide to switch to another.
- Disadvantages:
- 1. not easily measured, but whose effects might become apparent over the long run
- (2) have lost faith
- To combat credit card fraud, we developed a protocol or model for this suggested project. Most of the fundamental capabilities for identifying valid and fraudulent transactions may be supplied by this system. As a result of advancements in technology, it is becoming more difficult to monitor the habits and routines of those who engage in dishonest financial dealings. It is now possible to automate the process and save part of the effective amount of effort that is put into identifying credit card fraudulent activities due to the rise of machine learning, artificial intelligence, and other important disciplines of information technology.
- Excellent rate of detection.

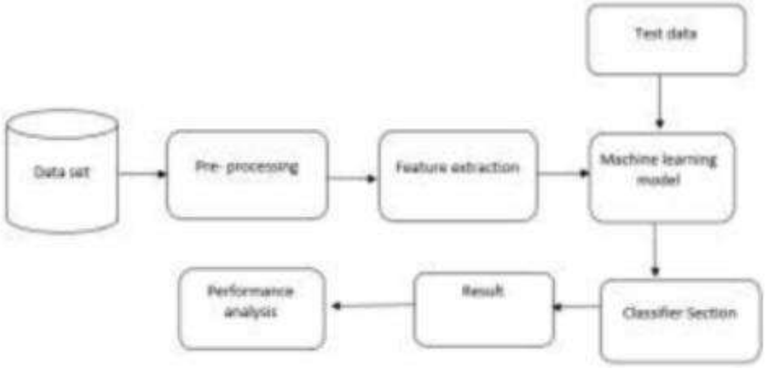


Fig.2: System architecture

MODULES:

1. **DATA COLLECTION:**

This article makes use of a database of product evaluations mined from customer credit card information. In this stage, you will narrow down the massive amount of accessible information to a manageable subset. To solve ML issues, you need data—ideally, a lot of data (examples or observations) for which you already know the desired result. Labeled data is information for which you already know the desired outcome. 7.1.2 SECOND MODULE: PREPROCESSING OF DATA

Get your data in order by formatting, cleaning, and sampling from the subset you've chosen. There are typically three phases of pre-processing data: Concerning the data's format, it's possible that it is not in a form easily used by you. Perhaps you have data stored in a relational database and would want to export it to a flat file, or perhaps you have data stored in a proprietary file format and would like to convert it to a relational database or a text file. When data is cleaned, it has errors corrected or missing information is removed. Some data instances may be missing information that you feel is necessary to solve the issue. It's possible that these occurrences should be eradicated. Some of the properties may include private information and may need to be scrubbed from the database. Sampling: You could find a plethora of data that you didn't know existed. Algorithms' execution durations, as well as their computational and memory needs, may become prohibitively lengthy when dealing with massive amounts of data. It may be more efficient to first evaluate just a subset of the data you've chosen, using that subset to explore and prototype potential solutions.

THE THIRD FEATURE EXTRACTION:

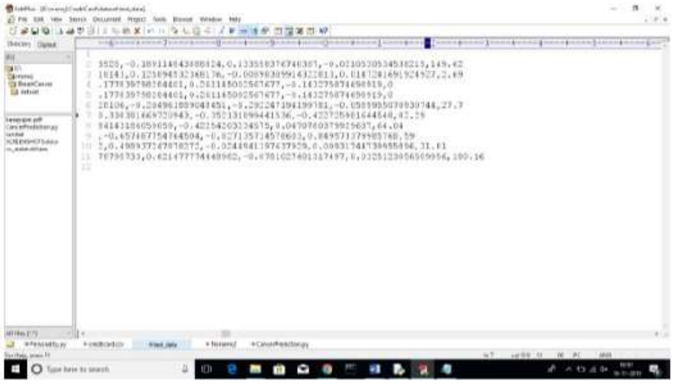
The next step, In order to streamline the process of collecting data, a procedure known as feature extraction is used. Feature extraction actually modifies the qualities, as opposed to feature selection which only ranks them in order of their predictive relevance. Transformed features are linear combinations of the original qualities. Classifier method is then used to train our models. Python's Natural Language Toolkit library's categorize module is used. We make use of the collected labeled dataset. The remaining portion of our labeled data will be utilized to evaluate the performance of the models. Classification of the preprocessed data was accomplished with the aid of several machine learning methods. Random forest classifiers were used. In text classification problems, these techniques are widely used.

4. Model for Evaluating Performance Model:

The process of testing and improving a model is fundamental to its creation. It helps in determining which model best reflects our data and how well that model will perform in the future. In data science, it is not acceptable to evaluate a model's efficacy using the same data that was used to train it. Doing so might lead to too optimistic and over-fitted results. Models in data science may be evaluated using either the Hold-Out or Cross-Validation technique. Overfitting is avoided in both approaches by comparing the model's results against data that it has never seen before (the test set). The average performance of the various categorization models is then calculated. The end product will be shown graphically. Using graphs to display information that has been previously categorised. Precision refers to how well the model performs on test data, measured by the proportion of its predictions that turn out to be right. Simply divide the number of right guesses by the total number of guesses to get the accuracy rate.

4. IMPLEMENTATION

For this task, we are use a credit card fraud dataset and a set of machine learning algorithms to identify potentially fraudulent transactions.



```
3528,-0.189114843888124,0.133518376740387,-0.0210570534538215,149.62
18141,0.125894532348176,-0.00898309914322813,0.0147281691924927,2.69
.177839798204401,0.241145092567677,-0.143275874698919,0
.177839798204401,0.241145092567677,-0.143275874698919,0
28106,-0.294961889043451,-0.292247394199781,-0.0589955070930744,27.7
3.334381469720943,-0.352131099441536,-0.422725981644548,42.28
5414318605965 9,-0.422542032345 75,0.0470760379925637,64.04
,-0.657467754764504,-0.827135714578603,0.869573379985768,59
2,0.498937247878275,-0.0244841197437929,0.0883174473995 5496,31.61
76795733,0.621477774448962,-0.678102740131 7497,0.0325123956509956,100.16
```

Fig.3: Dataset example

ALGORITHMS:

Random Forest Algorithm:

An accurate classifier model is constructed by using numerous classifier algorithms, since this is an ensemble algorithm. This approach uses a decision tree algorithm internally to create a trained model for classification.

The supervised machine learning algorithm random forest has widespread use in both classification and regression settings. For classification, it uses the sample with the highest vote count, and for regression, it uses an average or median of the samples. The ability of the Random Forest Algorithm to deal with data sets with both continuous and categorical variables, as in regression and classification, is one of its most notable properties. When used to classification difficulties, it produces superior outcomes.

An Algorithm for Logistic Regression:

One of the most well-known Machine Learning algorithms, logistic regression is a kind of Supervised Learning. The categorical dependent variable may be predicted from a collection of independent factors. The goal of logistic regression is to foretell the value of a dependent variable that may be classified. The result must so be a discrete or categorical number. There are two possible outcomes—"Yes" or "No," "0" or "1," "true" or "False," etc.—but the probability values between 0 and 1 are what are reported. In all but one respect, Logistic Regression and Linear Regression serve comparable purposes.

Calculus of Support Vector Machines:

Support Vector Machine, or SVM, is a common Supervised Learning technique for both classification and regression tasks. However, its primary use is in the realm of Machine Learning, where it is utilized for Classification tasks. To classify fresh data points efficiently in the future, the SVM algorithm seeks to find the optimal line or decision boundary that divides the space into n distinct classes. A hyperplane defines the optimal boundaries for making a choice.

The use of a decision tree algorithm:

In data mining, the decision tree is used for supervised learning purposes, namely in the areas of classification and regression. It's a decision-making aid in the form of a tree. Using a tree-like structure, the decision tree may provide either a classification or regression model. It breaks down a dataset into smaller chunks while gradually building up the decision tree. In the end, a tree containing both decision and leaf nodes is created. It takes at least two forks to make a decision at a node. An organization's decisions and categorizations are represented by the leaf nodes.

With the Xgboost algorithm, you may...

Python's XGBoost package implements gradient boosted decision trees with a focus on speed and execution, two of ML's most crucial requirements (machine learning). The academic community at the University of Washington introduced the Python module XgBoost (Extreme Gradient Boosting). It's a C++ add-on for Python that facilitates the training of ML model methods through Gradient Boosting.

5. EXPERIMENTAL RESULTS

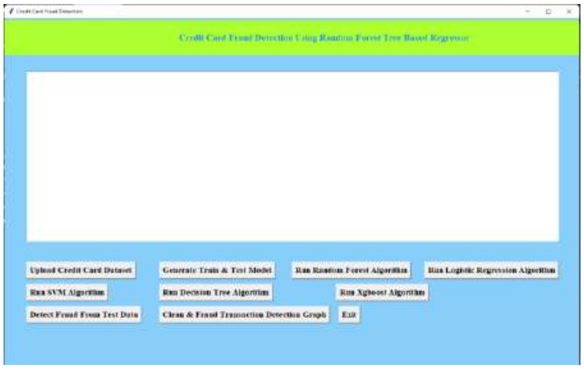
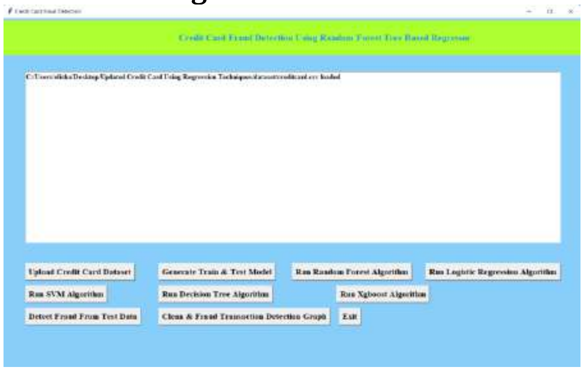
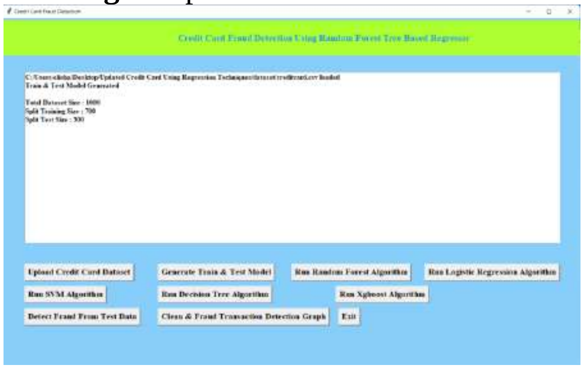

Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

Fig.4: Home screen


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

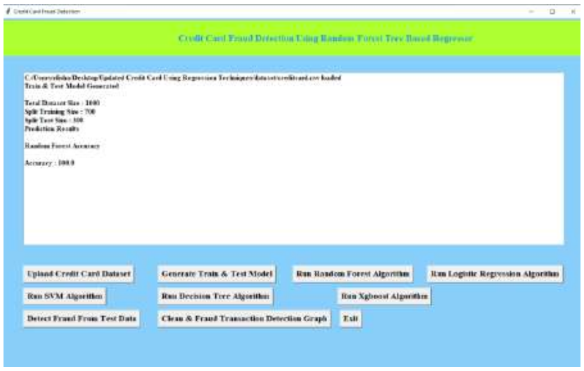
```
C:/Users/[illegible]/Desktop/Updated Credit Card Using Regression Techniques/dataset/creditcard.csv loaded
```

Fig.5: Upload credit card dataset


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

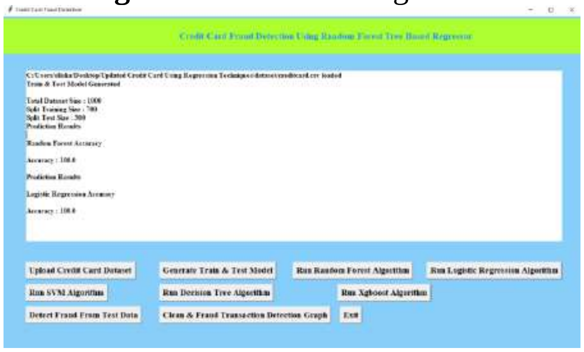
```
C:/Users/[illegible]/Desktop/Updated Credit Card Using Regression Techniques/dataset/creditcard.csv loaded
Train & Test Model Generated

Total Dataset Size : 1000
Split Training Size : 700
Split Test Size : 300
```

Fig.6: Generate train & test model


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

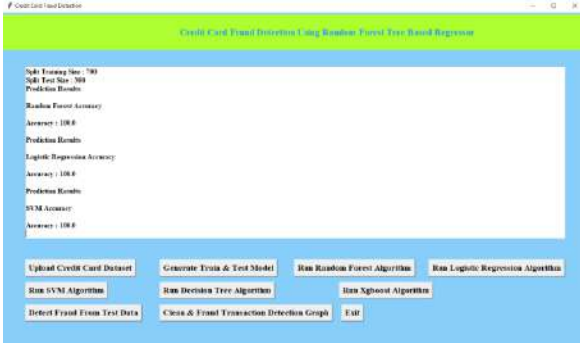
```
C:/Users/disha/Desktop/Updated Credit Card Using Regression Techniques/dataset/creditcard.csv loaded
Train & Test Model Generated

Total Dataset Size : 1000
Split Training Size : 700
Split Test Size : 300
Prediction Results

Random Forest Accuracy

Accuracy : 100.0
```

Fig.7: Random forest algorithm


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

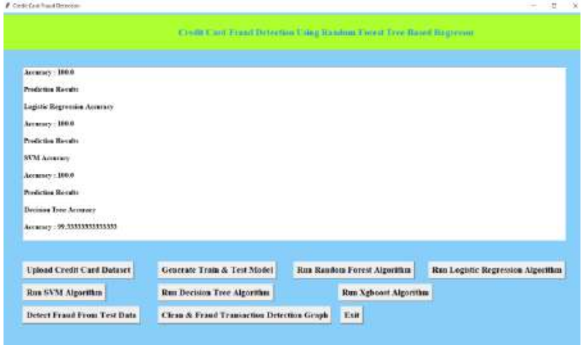
```
C:/Users/disha/Desktop/Updated Credit Card Using Regression Techniques/dataset/creditcard.csv loaded
Train & Test Model Generated

Total Dataset Size : 1000
Split Training Size : 700
Split Test Size : 300
Prediction Results

Random Forest Accuracy

Accuracy : 100.0

Prediction Results

Logistic Regression Accuracy

Accuracy : 100.0
```

Fig.8: Logistic regression algorithm


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

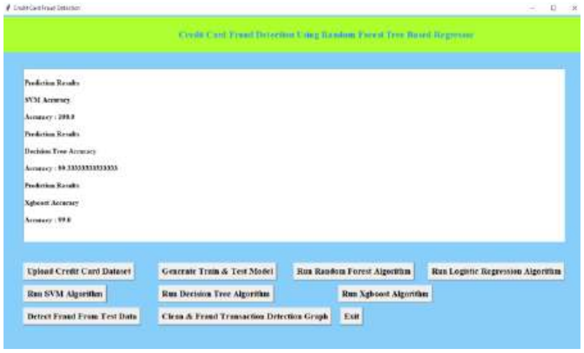
```
Split Training Size : 700
Split Test Size : 300
Prediction Results

Random Forest Accuracy

Accuracy : 100.0

Prediction Results

Logistic Regression Accuracy

Accuracy : 100.0

Prediction Results

SVM Accuracy

Accuracy : 100.0
```

Fig.9: SVM algorithm


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

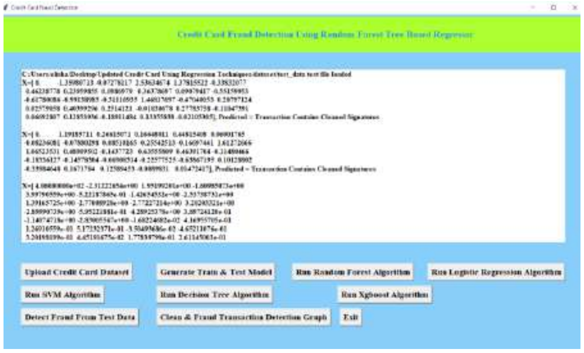
```
Accuracy : 100.0

Prediction Results

Logistic Regression Accuracy

Accuracy : 100.0

Prediction Results

SVM Accuracy

Accuracy : 100.0

Prediction Results

Decision Tree Accuracy

Accuracy : 99.33333333333333
```

Fig.10: Decision tree algorithm


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

```
Prediction Results
SVM Accuracy
Accuracy : 100.0
Prediction Results
Decision Tree Accuracy
Accuracy : 99.33333333333333
Prediction Results
Xgboost Accuracy
Accuracy : 99.0
```

Fig.11: Xgboost algorithm


Credit Card Fraud Detection
Credit Card Fraud Detection Using Random Forest Tree Based Regressor
Upload Credit Card Dataset
Generate Train & Test Model
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

```
C:/Users/alnha/Desktop/Updated Credit Card Using Regression Techniques/dataset/test_data test file loaded
X=[ 0.        -1.35980713 -0.07278117  2.53634674  1.37815522 -0.33832077
  0.46238778  0.23959855  0.0986979   0.36378697  0.09079417 -0.55159953
 -0.61780086 -0.99138985 -0.31116935  1.46817697 -0.47040053  0.20797124
  0.02579058  0.40399296  0.2514121  -0.01830678  0.27783758 -0.11047391
  0.06692807  0.12853936 -0.18911484  0.13355838 -0.02105305], Predicted = Transaction Contains Cleaned Signatures

X=[ 0.         1.19185711  0.26615071  0.16648011  0.44815408  0.06001765
 -0.08236081 -0.07880298  0.08510165 -0.25542513 -0.16697441  1.61272666
  1.06523531  0.48909502 -0.1437723   0.63555809  0.46391704 -0.11480466
 -0.18336127 -0.14578304 -0.06908314 -0.22577525 -0.63867195  0.10128802
 -0.33984648  0.1671704   0.12589453 -0.0089831   0.01472417], Predicted = Transaction Contains Cleaned Signatures

X=[ 4.06000000e+02 -2.31222654e+00  1.95199201e+00 -1.60985073e+00
  3.99790559e+00 -5.22187865e-01 -1.42654531e+00 -2.53738731e+00
  1.39165725e+00 -2.77008928e+00 -2.77227214e+00  3.20203321e+00
 -2.89990739e+00 -5.95221881e-01 -4.28925378e+00  3.89724120e-01
 -1.14074718e+00 -2.83005567e+00 -1.68224682e-02  4.16955705e-01
  1.26910559e-01  5.17232371e-01 -3.50493686e-02 -4.65211076e-01
  3.20198199e-01  4.45191675e-02  1.77839798e-01  2.61145003e-01
```

Fig.12: Detect fraud from test data

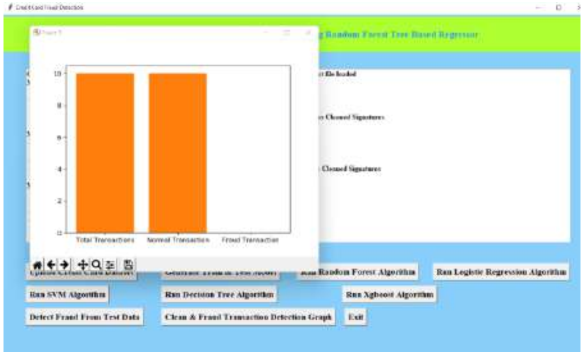

Credit Card Fraud Detection
Figure 1
g Random Forest Tree Based Regressor
10
8
6
4
2
0
Total Transactions
Normal Transaction
Fraud Transaction
t file loaded
n Cleaned Signatures
Cleaned Signatures
Run Random Forest Algorithm
Run Logistic Regression Algorithm
Run SVM Algorithm
Run Decision Tree Algorithm
Run Xgboost Algorithm
Detect Fraud From Test Data
Clean & Fraud Transaction Detection Graph
Exit

Fig.13: Clean & fraud transaction detection graph

6. CONCLUSION

With more training data, the Random forest algorithm can better predict outcomes, but it will take longer to verify and apply the results. Pre-processing procedures, if used more often, might also be beneficial. While the results demonstrated by other methods are impressive, they may have been much better if additional preprocessing had been done on the data

.

7. FUTURE WORK

We want to work on enhancing the model in the future by taking into account the properties of other dimensions beyond frequency, and then apply it to picture data.

REFERENCES

[1] X. Zhang, Y. Han, W. Xu, and Q. Wang, ``HOBA: A novel feature engineering methodology for credit card fraud detection with a deep learning architecture," Inf. Sci., May 2019. Accessed: Jan. 8, 2019.

[2] N. Carneiro, G. Figueira, and M. Costa, ``A data mining based system for credit-card fraud detection in e-tail," Decis. Support Syst., vol. 95, pp. 91101, Mar. 2017.

[3] B. Lebichot, Y.-A. Le Borgne, L. He-Guelton, F. Oblé, and G. Bontempi, ``Deep-learning domain adaptation techniques for credit cards fraud detection," in Proc. INNS Big Data Deep Learn. Conference, Genoa, Italy, 2019, pp. 7888.

[4] H. John and S. Naaz, ``Credit card fraud detection using local outlier factor and isolation forest," Int. J. Comput. Sci. Eng., vol. 7, no. 4, pp. 10601064, Sep. 2019.

[5] C. Phua, R. Gayler, V. Lee, and K. Smith-Miles, ``On the communal analysis suspicion scoring for identity crime in streaming credit applications," Eur. J. Oper. Res., vol. 195, no. 2, pp. 595612, Jun. 2009.

[6] Sudhamathy G: Credit Risk Analysis and Prediction Modelling of Bank Loans Using R, vol. 8, no-5, pp. 1954-1966.

[7] LI Changjian, HU Peng: Credit Risk Assessment for ural Credit Cooperatives based on Improved Neural Network, International Conference on Smart Grid and Electrical Automation vol. 60, no. - 3, pp 227-230, 2017.

[8] Wei Sun, Chen-Guang Yang, Jian-Xun Qi: Credit Risk Assessment in Commercial Banks Based On Support Vector Machines, vol.6, pp 2430-2433, 2006.

[9] Amlan Kundu, Suvasini Panigrahi, Shamik Sural, Senior Member, IEEE,“BLAST-SSAHA Hybridization for Credit Card Fraud Detection”, vol. 6, no. 4 pp. 309-315, 2009.

[10] Y. Sahin and E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, Proceedings of International Multi Conference of Engineers and Computer Scientists, vol. I, 2011.

[11] Sitaram patel, Sunita Gond , “Supervised Machine (SVM) Learning for Credit Card Fraud Detection, International of engineering trends and technology, vol. 8, no. -3, pp. 137- 140, 2014.

[12] Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, Rinku Badgujar," Credit Card Fraud Detection Using Decision Tree Induction Algorithm, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4, April-2015, pg. 92-95

[13] Dahee Choi and Kyungho Lee, “Machine Learning based Approach to Financial Fraud Detection Process in Mobile Payment System", vol. 5, no. - 4, December 2017, pp. 12-24.