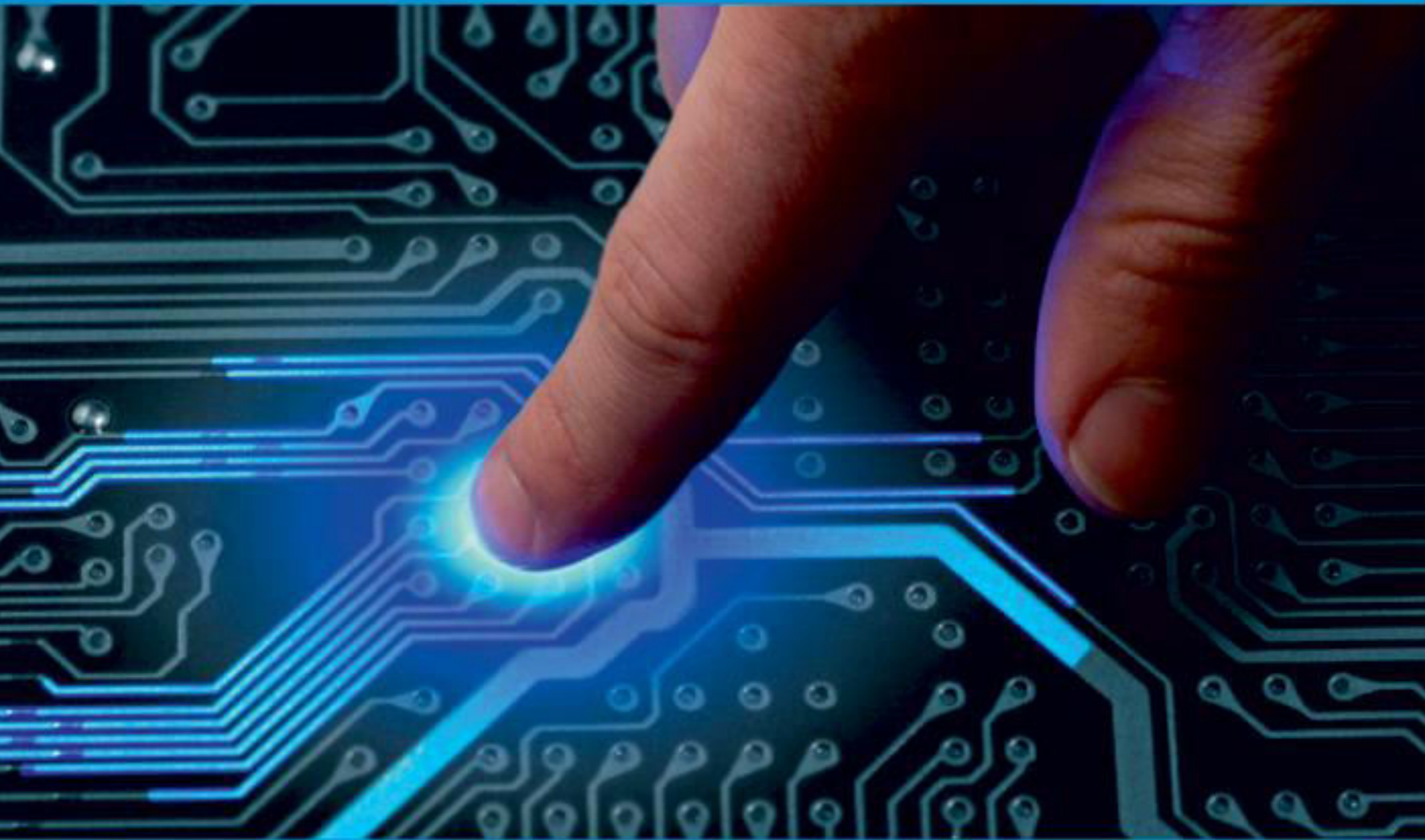
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 5, May 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005187|

Credit Card Fraud Detection System using Machine Learning

Pratik Chopade, Kanchan Patil, Shraddha Kalsekar, Atul Patil

Head of Department, Department of Computer Science, JSPM'S Rajashri Shahu College of Engineering, Pune, India

Guide, Department of Computer Science, JSPM'S Rajashri Shahu College of Engineering, Pune, India

Lecturer, Department of Computer Science, JSPM'S Rajashri Shahu College of Engineering, Pune, India

Student, Department of Computer Science, JSPM'S Rajashri Shahu College of Engineering, Pune, India

ABSTRACT: Credit card fraud detection is the process of identifying fraudulent purchasing attempts and rejecting them instead of processing the order. There are a variety of tools and strategies available to detect fraud, with many vendors using a few combinations of their own. The billions of plastic cards used worldwide are the gold mines of criminals. By 2027, financial services providers are expected to take $ 40 billion globally from credit card losses, a significant increase compared to $ 27.85 bn in 2018. This increase in losses is due in part to the increase in electronic sales. Imagine that today the average American has more than three credit cards, up to 1.5 billion cards in the US alone. While the number of plastic cards worldwide is estimated at 22.11 billion. Another reason is that fraudulent methods are becoming more complex and thus difficult to identify with standard fraud detection software.

KEYWORDS: Supervised learning, classification, regression, Logistic regression.

I. INTRODUCTION

While the e-commerce world has a lot to offer, one of the worst things about doing business online is dealing with credit card fraud. It is a source of income for every trader and trying to block it is a never-ending war. Card holders are often well caught up in fraudulent costs, especially in the US. All they have to do is tell their bank that the purchase is not authorized and unless there is clear evidence that they are lying - the costs will be deducted at no cost to them. Unfortunately, the cost of credit card fraud has to be somewhere, and in most cases, the seller ends up paying for it. In order to limit the amount of revenue lost to fraud, it is important for all traders to have effective measures to detect fraud. This usually means using a combination of different tools, from standard testing of card holder information to advanced risk detection algorithms. Let's take a look at some of the best ways to get credit card fraud online. Payment cards are easy to use because you only need to transfer a few simple bank numbers to identify your account and authorize transactions. This mention puts them at risk as well. It is very difficult to practice strong data security in a few simple numbers that should be shared with the organizations you work with. Credit card fraud costs the world economy more than $ 24 billion a year, and prices continue to rise. Small retailers are particularly vulnerable to the effects of fraud, which is why it is so important to have tools and procedures in place to detect fraud in your early stages.

II. REVIEWOF LITERATURE

As I have observed that, according to the FBI, credit card fraud is "the unauthorized use of a credit or debit card, or similar payment tool to fraudulently obtain money or property." All players involved in the card-based payment process can potentially fall victim to scammers, including:

- cardholders,
- online merchants,
- payment gateway providers,
- payment processing companies,
- credit card payment systems,
- card issuers (issuing banks), and
- acquirers (acquiring banks).

Except for cardholders whose anti-fraud measures narrow down to vigilance and timely reporting about lost or stolen cards, all other players rely on various digital tools designed to combat scams. The importance of these tools is hard to overstate. Say, if an online business shows a fraud rate greater than one percent, card networks like Mastercard or

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005187|

AmEx may cancel permission to accept and process credit card payments. With all the variety of fraudulent schemes involving credit cards, they can be roughly divided into two large groups - identity theft and transaction laundering.

III. ALGORITHMS

In this paper, we talk about Logistic Regression algorithms that are monitored by algorithms to detect fraudulent activity.

Logistic Regression:
Decreased performance is one of the most popular methods of machine learning, which comes under the supervision of a supervised learning strategy. It is used to predict phase-dependent fluctuations using a given set of independent variables.Depression predicts the outflow of phase-dependent variability. Therefore, the result should be phase or separate value. Either Yes or No, 0 or 1, True or False, etc. but instead of giving a direct value such as 0 and 1, it provides possible values between 0 and 1.Logistic Regression is very similar to Linear Regression regardless of how it is used. Linear Regression is used for troubleshooting problems, and Logistic regression is used for troubleshooting problems.In Logistic regression, instead of inserting a regression line, we are equal to the "S" shaped editing function, which predicts two higher values (0 or 1).A curve from a logistic activity indicates the possibility of something like cancer cells or not, the mouse is fat or not based on its weight, etc.Logistic Regression is an important machine learning algorithm because it has the ability to provide opportunities and separate new data using continuous and diverse data sets.

Logistic Regression Equation:
The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:
We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

IV. METHODOLOGY

We do analysis strategies for detecting user fraud friendly and secure. This program analyzes credit card fraud detection and proposes these adoption procedures and its evidence process. Contains only input numbers are the result of the PCA revolution. Unfortunately, due to privacy issues, we cannot provide real features and so on background information about data. Features V1, V2 ... V28 principal parts obtained by PCA, features only unchanged with PCA are 'Time' and 'Value'. The 'Time' feature contains seconds past between each action and first function in the database. Feature 'Value' The transaction value, this feature can be used as an example-depending on the cost reading. Feature 'Class' feedback it is variable and takes the wrong amount in the event of fraud and a good value in another way.

Implementation

Figure [1] Importing all required libraries

```
[ ] import numpy as np
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005187|

Figure [2] Loading Data

```
[ ]  # loading the dataset to a Pandas DataFrame
     credit_card_data = pd.read_csv('/content/credit_data.csv')
```

Figure [3] Data Understanding

```
[ ]  # first 5 rows of the dataset
     credit_card_data.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670

```
[ ]  credit_card_data.tail()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	4.356170	-1.593105	2.711941	-0.689256
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	-0.975926	-0.150189	0.915802	1.214756
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	-0.484782	0.411614	0.063119	-0.183699
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	-0.399126	-1.933849	-0.962886	-1.042082
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	-0.915427	-1.040458	-0.031513	-0.188093

Figure [4] Defining Data

```
# dataset informations
credit_card_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005187|

Figure [5] Inequality in data

```
[ ] # distribution of legit transactions & fraudulent transactions
    credit_card_data['Class'].value_counts()

    0    284315
    1       492
    Name: Class, dtype: int64
```

This Dataset is highly unblanced

0 --> Normal Transaction

1 --> fraudulent transaction

Figure [6] Print details of the Fraudulent transaction value

```
[ ] # separating the data for analysis
    legit = credit_card_data[credit_card_data.Class == 0]
    fraud = credit_card_data[credit_card_data.Class == 1]

[ ] print(legit.shape)
    print(fraud.shape)

    (284315, 31)
    (492, 31)
```

Figure [7] Bifurcation of Training and Test Data

```
[ ] model = LogisticRegression()

[ ] # training the Logistic Regression Model with Training Data
    model.fit(X_train, Y_train)

    LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                       intercept_scaling=1, l1_ratio=None, max_iter=100,
                       multi_class='auto', n_jobs=None, penalty='l2',
                       random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                       warm_start=False)
```

Model Evaluation

Accuracy Score

```
[ ] # accuracy on training data
    X_train_prediction = model.predict(X_train)
    training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy on Training data : ', training_data_accuracy)

    Accuracy on Training data :  0.9415501905972046

    # accuracy on test data
    X_test_prediction = model.predict(X_test)
    test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[ ] print('Accuracy score on Test Data : ', test_data_accuracy)

    Accuracy score on Test Data :  0.9390862944162437
```

V.CONCLUSION

Credit card fraud detection is an important research field. This is due to an increase in the number of fraud cases financial institutions. This issue opens the door to employment artificial intelligence to create systems that can detect fraud. Creating an AI-based system to detect fraud requires a database training system (or classifier). Data actually they are dirty and have poor numbers, noisy data, and foreign objects. Such problems adversely affect the level of system accuracy. To overcome these problems, logistic regression-based the separator is raised. Data is first cleaned using two methods: moderate and clustering-based method way. Second, the classifier is trained based on the verification process (wrap = 10), which ensures that everything The website is used both as a set of training data and test data set. Finally, the proposed separator is tested based on accuracy, sensitivity, and error rate metrics. Proposed a logistic regression-based classifier compared to a well-known one dividers, which is a group of neighbors close to K as well separating voting. Reversal phase based on order produces the best results (accuracy = 97.2%, sensitivity =97%, and error rate = 2.8%).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005187|

REFERENCES

[1] Yousefi, Niloofar, Marie Alaghband, and Ivan Garibay. "A Comprehensive Survey on Machine Learning Techniques and User Authentication Approaches for Credit Card Fraud Detection." arXiv preprint arXiv:1912.02629 (2019).

[2] Paschen, Jeannette, Jan Kietzmann, and Tim Christian Kietzmann. "Artificial intelligence (AI) and its implications for market knowledge in B2B marketing." Journal of Business & Industrial Marketing (2019).

[3] Abdallah, Aisha, MohdAizainiMaarof, and Anazida Zainal. "Fraud detection system: A survey." Journal of Network and Computer Applications 68 (2016): 90-113.

[4] Alladi, Tejasvi, et al. "Consumer IoT: Security vulnerability case studies and solutions." IEEE Consumer Electronics Magazine 9.2 (2020): 17-25.

[5] Rahman, Rizwan Ur, et al. "Classification of Spamming Attacks to Blogging Websites and Their Security Techniques." Encyclopedia of Criminal Activities and the Deep Web. IGI Global, 2020. 864-880.

[6] Somasundaram, Akila, and Srinivasulu Reddy. "Parallel and incremental credit card fraud detection model to handle concept drift and data imbalance." Neural Computing and Applications 31.1 (2019): 3-14.

[7] Gianini, Gabriele, et al. "Managing a pool of rules for credit card fraud detection by a Game Theory based approach." Future Generation Computer Systems 102 (2020): 549-561.

[8] Dal Pozzolo, Andrea, et al. "Credit card fraud detection: a realistic modeling and a novel learning strategy." IEEE transactions on neural networks and learning systems 29.8 (2017): 3784-3797.

[9] Wang, Chunhua, and Dong Han. "Credit card fraud forecasting model based on clustering analysis and integrated support vector machine." Cluster Computing 22.6 (2019): 13861-13866. [10] Deufel, Patrick, Jan Kemper, and Malte Brettel. "Pay now or pay later: A cross-cultural perspective on online payments." Journal of Electronic Commerce Research 20.3 (2019): 141-154.

INNO SPACE

SJIF Scientific Journal Impact Factor

Impact Factor: 8.165

doi crossref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING