Credit Card Fraud Detection Model

Abstract

The rapid evolution of technology has revolutionized the domain of payment methods, transitioning towards an online paradigm. Among these methods, credit cards have emerged as the most prevalent and efficient mode of payment, offering simplicity and swiftness in transactions. However, this transition has also brought about a surge in fraudulent activities, particularly in the realm of online credit card transactions. Such fraudulent activities have inflicted substantial financial losses on both financial institutions and consumers alike. In response to this pressing issue, this research endeavors to devise a robust model for detecting fraudulent credit card transactions utilizing machine learning algorithms implemented in Python. The project commences by acquiring credit card transaction data from the Kaggle platform, specifically reserved for research purposes of this nature. Subsequently, the acquired data undergoes preprocessing and filtering to render it amenable for analysis. The Logistic Regression algorithm is employed to train the model, enabling it to discern the distinctive patterns between legitimate and fraudulent transactions based on transaction parameters.

The model is trained using the Logistic Regression algorithm, enabling it to differentiate between legitimate and fraudulent transactions based on transaction parameters. To assess the model's effectiveness, a separate set of test data, unseen during training, is used to evaluate its accuracy in identifying fraudulent transactions. The model achieves an accuracy of 0.9987, indicating its success in detecting previously unseen transactions.Furthermore, the accuracy of the test data is thoroughly examined to comprehensively evaluate the model's performance, yielding a similar accuracy of 0.998, indicating the model's proficiency in handling new data.Finally, the findings are visually represented through bar charts, elucidating the accuracy of both the training and test data sets. Furthermore, a graphical depiction of the distribution of legitimate and fraudulent data within the test datasets is presented, providing insights into the model's detection capabilities. Through the amalgamation of advanced machine learning techniques and Python programming, this research contributes to the ongoing efforts in fortifying the security measures surrounding online credit card transactions, thereby mitigating the detrimental impact of fraudulent activities on financial stakeholders and consumers.

I. Introduction

Credit card fraud poses an increasingly pressing and disconcerting challenge in our rapidly evolving technological landscape. As technology advances at an unprecedented pace, security measures must equally evolve and fortify themselves against imminent threats. The surges in digital payment methods and government-driven initiatives promoting the use of plastic money have exacerbated the complexity of this issue[1].

Credit card fraud, defined as the unauthorized use of another person's credit card or credit card information for fraudulent transactions, inflicts substantial financial losses upon both victims and credit card companies. In response to this pervasive issue, credit card fraud detection systems have become an indispensable component of the credit card industry[2].

Credit card fraud can manifest through various methods, including skimming, phishing, counterfeiting, and identity theft. Among these, skimming ranks among the most prevalent, wherein fraudsters employ small devices known as skimmers to illicitly obtain credit card details from unsuspecting individuals. Phishing is another common tactic, involving deceptive emails or websites designed to trick victims into disclosing their credit card information[3].

Traditional credit card fraud detection software employs diverse techniques, including pattern recognition, anomaly detection, and predictive modeling. While these systems analyze copious transaction data, they often fall short in identifying cunning fraud attempts that may appear innocuous on the surface but pose significant financial risks internally. Consequently, conventional fraud detection systems struggle to efficiently detect sophisticated fraud schemes due to inherent limitations in their design and functionality[4].

To address this challenge, our project adopts machine learning algorithms, leveraging Python libraries such as Pandas, NumPy, MatLab, and MatPlotLib for data analysis and visualization.Linear Regression Machine learning algorithms is employed in this study, the algorithm possess the unique capability to scrutinize vast datasets and unveil patterns indicative of illicit behavior, making them highly adept at uncovering credit card fraud. They excel in spotting transaction irregularities, such as those occurring at unusual times or locations and involving atypical amounts, which often elude traditional detection methods[5].

In essence, our endeavor aims to join the power of machine learning to develop a robust and accurate fraud detection system, ensuring the security of financial transactions in an ever-evolving digital landscape.

II. Statement of the Problem and justification

Because of advancements in e-commerce systems and communication technology, credit cards have emerged as one of the most prevalent payment methods for both every day and online transactions. Unfortunately, this widespread adoption has led to a significant surge in associated fraud. Each year, illicit credit card transactions result in substantial losses for both businesses and individuals. Fraudsters have adeptly leveraged technology to siphon funds from unsuspecting victims, necessitating a proactive response to thwart their malicious activities.

When a credit card is duplicated or stolen, the ensuing transactions are classified as fraudulent. Detecting and preventing these illicit transactions in a timely manner is of utmost importance, as the resultant financial losses can be substantial. With the increasing ubiquity of credit card usage, the financial toll inflicted by credit card fraud continues to mount. Simultaneously, fraudsters continually explore new technological avenues to perpetrate their illicit schemes.

Hence, the objective of this study is to devise a precise model for detecting fraudulent credit card transactions employing logistic algorithms. By doing so, we aim to address the aforementioned issues and mitigate the impact of credit card fraud on both individuals and businesses.

III. Literature Review

This chapter delves into the existing body of knowledge surrounding credit card fraud detection model, offering a concise synthesis of key findings and insights from previous research. Through this exploration, this study is aim to contextualize within the broader scholarly discourse and identify gaps for further investigation[6].

a. What is Credit Card Fraud?

Credit card fraud can be defined as the intentional and unauthorized manipulation or exploitation of credit card information, payment mechanisms, or transactional processes, undertaken with the aim of illicitly acquiring financial gain or benefits at the expense of legitimate cardholders, financial institutions, or merchants[7]. It encompasses a spectrum of deceptive practices, including but not limited to identity theft, card-present fraud, and card-not-present fraud, often facilitated by

sophisticated techniques such as phishing, skimming, or data breaches. At its core, credit card fraud represents a breach of trust and integrity within the financial ecosystem, posing significant economic, regulatory, and social challenges that necessitate proactive detection, prevention, and mitigation strategies to safeguard against its deleterious effects[8].

b. Types of Credit Card Fraud

Understanding distinct types of credit card fraud is essential within academic research contexts, serving as a foundational framework for the development of robust prevention and detection strategies aimed at mitigating financial losses and safeguarding stakeholders against fraudulent activities.

i. **Identity Theft:** Identity theft is a prevalent form of credit card fraud characterized by the illicit acquisition and misuse of personal information, including but not limited to Social Security numbers, addresses, and dates of birth. Perpetrators often employ techniques such as phishing, hacking, or exploiting data breaches to obtain sensitive data, subsequently utilizing it to open fraudulent credit card accounts or conduct unauthorized transactions[9].

ii. **Card-Present Fraud:** This category of credit card fraud transpires when stolen or counterfeit credit cards are physically presented during transactions, typically at point-of-sale terminals. Perpetrators may utilize various means to acquire stolen cards, fabricate counterfeit ones, or manipulate card data, thereby facilitating unauthorized purchases and transactions[10].

iii. **Card-Not-Present Fraud:** In contrast to card-present fraud, card-not-present fraud unfolds in remote transactions where the physical card is not required, such as online or telephone purchases. Here, fraudsters exploit compromised credit card information obtained through diverse methods to execute unauthorized transactions without the cardholder's knowledge or consent[9].

iv. **Account Takeover:** In an account takeover, fraudsters gain unauthorized access to a legitimate cardholder's account through various means such as phishing, social engineering, or hacking. Once access is obtained, they may change account details, make unauthorized transactions, or transfer funds to other accounts[11].

v. **Friendly Fraud:** Also known as chargeback fraud, friendly fraud occurs when a legitimate cardholder disputes a valid transaction with their bank or credit card issuer, often claiming that they did not authorize or receive the goods or services purchased. This type of fraud can result in financial losses for merchants and issuers[12].

vi. **Application Fraud:** Application fraud involves using false or stolen information to apply for a credit card or line of credit. Fraudsters may fabricate identities or use stolen identities to obtain credit cards, which they then use for unauthorized transactions before defaulting on payments[13].

vii. **Skimming**: involves the illegal capture of credit card information by installing hidden devices (skimmers) on legitimate card readers, such as ATMs or point-of-sale terminals. These devices capture card details, which are then used to create counterfeit cards or make unauthorized transactions[14].

c. Overview of Logistic Regression Machine Learning Algorithm

Logistic regression is a statistical modeling technique used for binary classification tasks, where the goal is to predict the probability of an observation belonging to one of two possible categories or classes. Unlike linear regression, which predicts continuous outcomes, logistic regression models the probability using the logistic function, also known as the sigmoid function. The output of logistic regression is constrained between 0 and 1, representing the probability of the event (e.g., Legitimate Transaction, Fraudulent Transaction,) occurring[5].

d. Why Employing Logistic Regression in Credit Card Fraud Detection?

i. In the context of credit card fraud detection, logistic regression is a statistical modeling technique used to predict the likelihood that a given credit card transaction is fraudulent based on various features associated with the transaction. These features typically include transaction amount, location, time of day, type of merchant, previous transaction history, and more[10].

ii. Logistic regression is a type of binary classification algorithm that estimates the probability that an observation belongs to one of two classes; fraudulent or legitimate transactions. Unlike linear regression, which predicts continuous outcomes, logistic regression models the probability using the logistic function, also known as the sigmoid function. The output of logistic regression is constrained between 0 and 1, representing the probability of the event (fraud) occurring[15].

iii. In credit card fraud detection, logistic regression works by learning a decision boundary that separates fraudulent transactions from legitimate ones based on the provided features. During the training phase, the logistic regression algorithm adjusts the parameters (coefficients) of the

model to maximize the likelihood of the observed fraudulent transactions while minimizing the likelihood of legitimate transactions being misclassified as fraudulent[16].

iv. Once trained, the logistic regression model can then be used to predict the probability of fraud for new, unseen transactions. By setting a threshold probability, financial institutions can classify transactions as either fraudulent or legitimate based on whether the predicted probability exceeds the threshold[17].

e. Overview of Python, Jupyter Notebook, Pandas and other libraries

In the realm of credit card fraud detection, Python, Jupyter Notebook, Pandas, and other associated libraries play pivotal roles in facilitating robust analysis and effective detection strategies.

i. Python, with its simplicity, versatility, and extensive ecosystem of libraries, emerges as a preferred choice for developing fraud detection systems. Its readability and vast community support make it ideal for rapid prototyping and deployment of machine learning models[18].

ii. Jupyter Notebook serves as a dynamic platform for interactive data exploration, visualization, and model development. Its integration with Python enables researchers and analysts to iteratively explore datasets, experiment with algorithms, and visualize results seamlessly, fostering a more intuitive and efficient workflow[19].

iii. Pandas, a powerful data manipulation and analysis library, empowers practitioners to preprocess and transform raw transaction data efficiently. Its intuitive data structures and functions facilitate tasks such as data cleaning, feature engineering, and statistical analysis, laying the foundation for building accurate fraud detection models[18].

iv. Furthermore, other libraries such as NumPy, SciPy, Scikit-learn, and Matplotlib complement Python and Pandas, enriching the fraud detection pipeline with advanced numerical computing, machine learning algorithms, and visualization capabilities[20].

In summary, Python, Jupyter Notebook, Pandas, and associated libraries form a robust ecosystem for credit card fraud detection, enabling researchers and analysts to leverage the power of data science and machine learning to combat fraudulent activities effectively.

f. Review of the related literature

Several recent studies conducted by various scholars have contributed to the field of credit card fraud detection. These studies have explored different algorithms and methodologies to address the growing problem of fraudulent transactions.

Here is a summary of some of these studies:

i. Renuka Devi has addressed the pressing issue of credit card fraud in the digital era. They assert that the proliferation of online payments and the heightened reliance on credit cards post-pandemic have exacerbated the challenge of fraud detection. Traditional fraud detection mechanisms, they argue, are hampered by inherent limitations, particularly in identifying sophisticated fraudulent activities. In response to this, Devi and Ray propose a credit card fraud detection model leveraging machine learning and convolutional neural networks, recognized for their efficacy in predictive analysis. Their model integrates simple yet potent technologies to ensure robust and accurate fraud detection, encompassing techniques such as pattern recognition, anomaly detection, and predictive modeling. However, they caution that while these methods are commonly utilized in fraud detection software, they may inadvertently overlook subtle fraudulent transactions, thereby exposing organizations to significant financial risks. However, the proposed model, as outlined by Devi and Ray, entails preprocessing techniques, weighted average calculations, and training utilizing machine learning algorithms like Logistic Regression, SVM, and K-Nearest Neighbor. Nonetheless, they acknowledge that the complexity of data preprocessing techniques discussed in their paper, including outlier rectification and feature extraction, may present challenges in practical implementation, particularly for users with limited technical expertise. Furthermore, Devi and Ray highlight that the implementation and maintenance of the AI/ML/CNN model for fraud detection could necessitate substantial computational resources and expertise, potentially rendering it less accessible for smaller financial institutions or organizations with constrained resources[21].

ii. The research paper by Kolli Nikhil et al. proposes a CatBoost-based system for detecting credit card fraud, with the aim of accurately identifying fraudulent transactions while minimizing false positives to maintain customer satisfaction. CatBoost, a machine learning algorithm, is highlighted for its proficiency in handling categorical features and unbalanced datasets, rendering it suitable for credit card fraud detection.The evaluation of the model's efficiency in spotting fraudulent transactions is conducted using various performance indicators such as precision, recall, and F1-score. The paper underscores the importance of robust credit card fraud detection models in light of the significant financial losses associated with credit card theft, emphasizing the potential of machine learning algorithms like CatBoost in effectively addressing this issue.However, the research paper does not explicitly discuss specific limitations or challenges encountered during the implementation or evaluation of the CatBoost-based credit card fraud detection system. While it acknowledges the effectiveness of CatBoost in managing categorical

features and unbalanced datasets, it does not delve into potential drawbacks or areas where the algorithm may not perform optimally. The focus of the paper is primarily on presenting the proposed CatBoost-based system for credit card fraud detection and evaluating its efficiency using performance indicators, without discussing potential limitations or areas for improvement in the model.Furthermore, the paper does not address any external factors or real-world constraints that could impact the practical implementation of the proposed fraud detection system using CatBoost. Overall, the limitations of the research paper lie in the lack of discussion on specific challenges faced during the study, potential drawbacks of the CatBoost algorithm in this context, and considerations for real-world application and scalability of the proposed system[22].

iii. The study conducted by Dhwanir Shah and Lokesh Kumar Sharma emphasizes the importance of implementing a secure credit card fraud detection system to mitigate financial losses stemming from fraudulent transactions. Notably, Decision trees and Random Forest algorithms are singled out for their efficacy in dataset analysis and accurate identification of fraudulent transactions.In their research, Shah and Sharma employ data preprocessing techniques such as OneHotEncoding and Target Guided Mean encoding to optimize the dataset for classification tasks. They present a performance evaluation of the Decision Tree classifier, both before and after parameter tuning, using confusion matrices to demonstrate the model's enhanced accuracy and effectiveness in fraud detection.However, it is noted that the dataset utilized in the study is simulated, potentially lacking the complexity and variability of real-world credit card transaction data. Shah and Sharma acknowledge that this simulated dataset may result in classifiers achieving 100% accuracy, which might not accurately reflect their performance in a more realistic setting. Additionally, the paper does not extensively delve into the computational complexity or scalability of the proposed fraud detection system, aspects crucial for real-time applications or handling large-scale datasets. Furthermore, the evaluation metrics employed to gauge the models' performance are not thoroughly discussed, potentially limiting the comprehensive understanding of the model's effectiveness beyond accuracy[23].

iv. Sandhya et al. discussed the application of machine learning techniques in credit card fraud detection. They evaluated algorithms including Naive Bayes, Bernoulli, and Random Forest, focusing on metrics such as accuracy, recall, and F1-score. The study demonstrated the efficacy of these algorithms in analyzing customer transaction data streams for detecting fraudulent activities, with Random Forest exhibiting superior performance in accuracy and precision for fraud detection. The classification report, delineating class 0 as valid transactions and class 1 as

fraudulent transactions, was provided. Moreover, the authors highlighted the limitations of using accuracy from the confusion matrix for unbalanced categorization, proposing computation of accuracy score and precision by comparing false positives generated by the code to actual occurrences[24].

v. The study conducted by Varun Kumar K S et al employed various machine learning algorithms for fraud detection. Logistic Regression was used for classification, Decision Trees for both classification and regression, and K Nearest Neighbor (KNN) algorithm was explored as well. Logistic Regression was supplemented with synthetic minority oversampling to handle data imbalance. However, the paper lacked in-depth discussions on crucial aspects such as dealing with skewed data, class imbalance, and handling categorical data in fraud detection, which are vital in real-world scenarios. Additionally, there was limited exploration on the interpretability of the models, scalability, computational complexity, generalizability to different datasets, adaptability to evolving fraud patterns, and potential drawbacks of the techniques used, which are all essential considerations for deploying effective fraud detection systems[3].

IV. Methodology

This section delineates the methodology utilized throughout the study. It elucidates the study's design, defines the population sample, and outlines the data collection methods including the instruments employed and the procedural steps undertaken. Additionally, it delves into the methods employed for data analysis

a. Research design

The methodology adopted for this research was qualitative in nature, chosen specifically to delve into and comprehend the intricacies of the dataset. Given the primary objective of scrutinizing and distinguishing between legitimate and fraudulent transactions, a qualitative approach emerged as the most fitting strategy[25].

Qualitative research was deemed essential as it allowed for a nuanced exploration of the dataset's parameters, facilitating a deeper understanding of the underlying patterns and anomalies within[26]. By immersing ourselves in the data, we were able to discern subtle nuances that may have eluded a purely quantitative analysis.

Moreover, the complexity of the task at hand left us with no alternative but to employ qualitative methods. Unlike quantitative techniques, which primarily focus on numerical data and statistical

analysis, qualitative research offered the flexibility to probe into the contextual nuances and subjective factors that often play a pivotal role in discerning fraudulent activities[27].

In essence, the decision to utilize qualitative research methodology was driven by the need to explore the multifaceted nature of the dataset comprehensively. By adopting this approach, we were able to gain deeper insights into the dynamics of legitimate and fraudulent transactions, thereby enhancing the effectiveness of our analysis and decision-making processes.

Research Architecture

The Research Architecture depicted below illustrates the sequential flow of the research process. It commences with data acquisition from the Kaggle website, focusing on transactional data earmarked specifically for detecting credit card fraud. Following data collection, the next phase involves data preprocessing, aimed at cleansing and formatting the data to facilitate analysis and modeling[28]. Subsequently, data analysis ensues, involving statistical manipulations and calculations to inform the development of a logistic regression model.

Further along, the dataset undergoes division into training and testing subsets, crucial for model development through iterative training and evaluation. Here, logistic regression is employed to train the model using the training data, distinguishing between legitimate and fraudulent transactions[29]. Finally, the model undergoes evaluation by testing its performance on the unseen test dataset, providing insights into its efficacy and robustness.

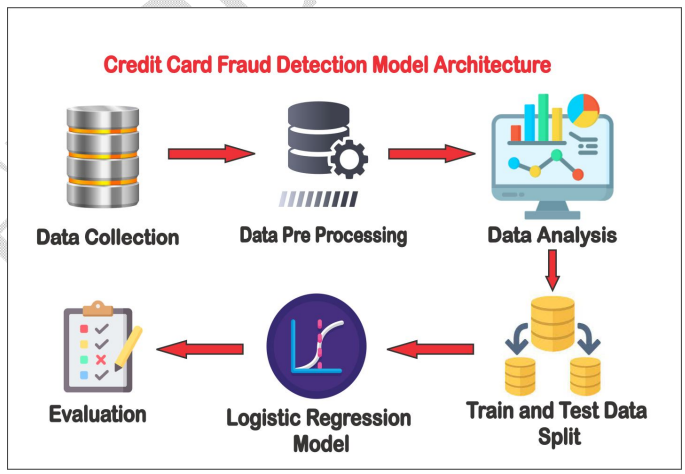

Figure 1- Research Architecture

b. Data collection method

For this research, the data collection method employed is web scraping, involving the direct retrieval of data from third-party websites[30]. Specifically, data was sourced from the Kaggle website using Jupyter Notebook in Python. The dataset obtained comprises credit card transactions made by

European cardholders in the year 2023. With over 550,000 records, the dataset has undergone anonymization (refer to the process of removing or altering personally identifiable information from data sets, thus making it anonymous) to safeguard the identities of the cardholders. The main purpose of utilizing this dataset is to support the development of algorithms and models aimed at detecting fraudulent transactions effectively. By leveraging this dataset, researchers can explore patterns and characteristics indicative of potential fraud, thus enhancing the accuracy and efficiency of fraud detection mechanisms[31].

The following screenshot depicts a Jupyter Notebook displaying Python code used to download a dataset from the Kaggle website. Notably, the variable 'dataset' contains the web address where the dataset is located on the Kaggle website. Later, the 'opendatasets' library, imported as 'od', is utilized to directly download the dataset to the specified location.

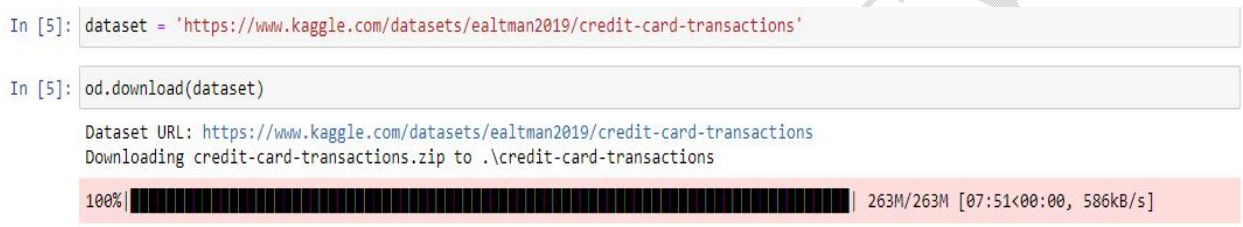

Figure 2- Downloading the Datasets

c. Sampling technique

This research utilizes the Credit Card Transactions Dataset from Kaggle, comprising a comprehensive record of credit card transactions over a specified period, including both genuine and fraudulent transactions. With a total of 568,630 entries, it's essential to balance the dataset, ensuring equal representation of fraudulent and legitimate transactions, resulting in 284,315 instances for each. The data is then divided into four categories: X_train: This contains the features (input variables) used for training the machine learning model. Features could include things like transaction amount, location, time of transaction, etc. Y_train: This contains the corresponding labels or target values for the training set. In this case, it would indicate whether each transaction is fraudulent or not. Typically, 0 might indicate a legitimate transaction, while 1 might indicate a fraudulent one. X_test: This contains a separate set of features that are used for testing the trained model's performance. It's important that the model doesn't see this data during training, as it should evaluate how well it generalizes to unseen data. Y_test: This contains the corresponding labels or target values for the test

set. Like Y_train, it indicates whether each transaction is fraudulent or not, but it's used for evaluating the model's performance[32].

The table below presents the count of legitimate transactions and fraudulent transactions, both totaling 284,315. Each dataset consists of 31 columns.

```
print(Legit_Transaction.shape)
print (Fraud_Transaction.shape)
```

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

Figure 3- Legit and Fraud Transaction Sampling

Likewise, the table below illustrates the sampled data for X_train, X_test, Y_train, and Y_test, utilized in training and testing the logistic regression model.

```
print (X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(454904, 30) (113726, 30) (454904,) (113726,)
```

Figure 4- Sampling for Training and Test Datasets

d. Data Pre Processing

Before analysis and visualization, it is essential to preprocess a dataset to align it with a usable pattern. This includes checking for any null values in rows or columns and ensuring a balance between the occurrences of legitimate and fraudulent transactions[33]. In this study, the dataset underwent the following preprocessing steps prior to analysis:

i. To start, the first five rows and the last five rows of the data are displayed to provide an overview of the dataset, aiding in understanding its structure and other key parameters. This can be observed in the screenshot of the Jupyter Notebook below.

```
In [11]: print (credit_dt.head())

   id        V1        V2        V3        V4        V5        V6        V7  \
0   0 -0.260648 -0.469648  2.496266 -0.083724  0.129681  0.732898  0.519014
1   1  0.985100 -0.356045  0.558056 -0.429654  0.277140  0.428605  0.406466
2   2 -0.260272 -0.949385  1.728538 -0.457986  0.074062  1.419481  0.743511
3   3 -0.152152 -0.508959  1.746840 -1.090178  0.249486  1.143312  0.518269
4   4 -0.206820 -0.165280  1.527053 -0.448293  0.106125  0.530549  0.658849

         V8        V9  ...       V21       V22       V23       V24       V25  \
0 -0.130006  0.727159  ... -0.110552  0.217606 -0.134794  0.165959  0.126280
1 -0.133118  0.347452  ... -0.194936 -0.605761  0.079469 -0.577395  0.190090
2 -0.095576 -0.261297  ... -0.005020  0.702906  0.945045 -1.154666 -0.605564
3 -0.065130 -0.205698  ... -0.146927 -0.038212 -0.214048 -1.893131  1.003963
4 -0.212660  1.049921  ... -0.106984  0.729727 -0.161666  0.312561 -0.414116

        V26       V27       V28    Amount  Class
0 -0.434824 -0.081230 -0.151045  17982.10      0
1  0.296503 -0.248052 -0.064512   6531.37      0
2 -0.312895 -0.300258 -0.244718   2513.54      0
3 -0.515950 -0.165316  0.048424   5384.44      0
4  1.071126  0.023712  0.419117  14278.97      0

[5 rows x 31 columns]

In [12]: print (credit_dt.tail())

             id        V1        V2        V3        V4        V5        V6  \
568625  568625 -0.833437  0.061886 -0.899794  0.904227 -1.002401  0.481454
568626  568626 -0.670459 -0.202896 -0.068129 -0.267328 -0.133660  0.237148
568627  568627 -0.311997 -0.004095  0.137526 -0.035893 -0.042291  0.121098
568628  568628  0.636871 -0.516970 -0.300889 -0.144480  0.131042 -0.294148
568629  568629 -0.795144  0.433236 -0.649140  0.374732 -0.244976 -0.603493

             V7        V8        V9  ...       V21       V22       V23  \
568625 -0.370393  0.189694 -0.938153  ...  0.167503  0.419731  1.288249
568626 -0.016935 -0.147733  0.483894  ...  0.031874  0.388161 -0.154257
568627 -0.070958 -0.019997 -0.122048  ...  0.140788  0.536523 -0.211100
568628  0.580568 -0.207723  0.893527  ... -0.060381 -0.195609 -0.175488
568629 -0.347613 -0.340814  0.253971  ...  0.534853 -0.291514  0.157303

             V24       V25       V26       V27       V28    Amount  Class
568625 -0.900861  0.560661 -0.006018  3.308968  0.081564   4394.16      1
568626 -0.846452 -0.153443  1.961398 -1.528642  1.704306   4653.40      1
568627 -0.448909  0.540073 -0.755836 -0.487540 -0.268741  23572.85      1
568628 -0.554643 -0.099669 -1.434931 -0.159269 -0.076251  10160.83      1
568629  0.931030 -0.349423 -1.090974 -1.575113  0.722936  21493.92      1

[5 rows x 31 columns]
```

Figure 5- The Head and the Tail of the Datasets

ii. The dataset undergoes further scrutiny using the Isnull function to detect any missing values that could potentially compromise the accuracy of the results. The screenshot below confirms that the dataset contains no null values.

```
In [13]: credit_dt.isnull

Out[13]: <bound method DataFrame.isnull of            id        V1        V2        V3        V4        V5        V6  \
0           0 -0.260648 -0.469648  2.496266 -0.083724  0.129681  0.732898
1           1  0.985100 -0.356045  0.558056 -0.429654  0.277140  0.428605
2           2 -0.260272 -0.949385  1.728538 -0.457986  0.074062  1.419481
3           3 -0.152152 -0.508959  1.746840 -1.090178  0.249486  1.143312
4           4 -0.206820 -0.165280  1.527053 -0.448293  0.106125  0.530549
...       ...       ...       ...       ...       ...       ...       ...
568625  568625 -0.833437  0.061886 -0.899794  0.904227 -1.002401  0.481454
568626  568626 -0.670459 -0.202896 -0.068129 -0.267328 -0.133660  0.237148
568627  568627 -0.311997 -0.004095  0.137526 -0.035893 -0.042291  0.121098
568628  568628  0.636871 -0.516970 -0.300889 -0.144480  0.131042 -0.294148
568629  568629 -0.795144  0.433236 -0.649140  0.374732 -0.244976 -0.603493

             V7        V8        V9  ...       V21       V22       V23  \
0      0.519014 -0.130006  0.727159  ... -0.110552  0.217606 -0.134794
1      0.406466 -0.133118  0.347452  ... -0.194936 -0.605761  0.079469
2      0.743511 -0.095576 -0.261297  ... -0.005020  0.702906  0.945045
3      0.518269 -0.065130 -0.205698  ... -0.146927 -0.038212 -0.214048
4      0.658849 -0.212660  1.049921  ... -0.106984  0.729727 -0.161666
...         ...       ...       ...  ...       ...       ...       ...
568625 -0.370393  0.189694 -0.938153  ...  0.167503  0.419731  1.288249
568626 -0.016935 -0.147733  0.483894  ...  0.031874  0.388161 -0.154257
568627 -0.070958 -0.019997 -0.122048  ...  0.140788  0.536523 -0.211100
568628  0.580568 -0.207723  0.893527  ... -0.060381 -0.195609 -0.175488
568629 -0.347613 -0.340814  0.253971  ...  0.534853 -0.291514  0.157303

            V24       V25       V26       V27       V28    Amount  Class
0      0.165959  0.126280 -0.434824 -0.081230 -0.151045  17982.10      0
1     -0.577395  0.190090  0.296503 -0.248052 -0.064512   6531.37      0
2     -1.154666 -0.605564 -0.312895 -0.300258 -0.244718   2513.54      0
3     -1.893131  1.003963 -0.515950 -0.165316  0.048424   5384.44      0
4      0.312561 -0.414116  1.071126  0.023712  0.419117  14278.97      0
...         ...       ...       ...       ...       ...       ...    ...
568625 -0.900861  0.560661 -0.006018  3.308968  0.081564   4394.16      1
568626 -0.846452 -0.153443  1.961398 -1.528642  1.704306   4653.40      1
568627 -0.448909  0.540073 -0.755836 -0.487540 -0.268741  23572.85      1
568628 -0.554643 -0.099669 -1.434931 -0.159269 -0.076251  10160.83      1
568629  0.931030 -0.349423 -1.090974 -1.575113  0.722936  21493.92      1

[568630 rows x 31 columns]>
```

Figure 6- Checking the empty values

iii. The data is additionally processed by segregating it into two variables: the first variable stores records of legitimate transactions, defined as transactions with a class equal to 0, while the second variable stores fraudulent transactions, identified by a class equal to 1. Following this segmentation, the frequency of each occurrence and its corresponding column number are retrieved.

```
In [44]: Legit_Transaction = credit_dt [credit_dt.Class==0]
         Fraud_Transaction =  credit_dt [credit_dt.Class==1]

In [45]: print(Legit_Transaction.shape)
         print (Fraud_Transaction.shape)

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

Figure 7- Variable for Legit and Fraud Transactions

v. Here, a new variable named "new_dataset" is created to accommodate the concatenated sample of legitimate and fraudulent transactions. This new dataset is utilized from this stage onward for easier access to the datasets. It's evident from the screenshot below that the values of each occurrence in the balance have an equal number, ensuring accurate results.

```
In [24]: Fraud_Sample = Fraud_Transaction.sample(n=284315)

In [25]: new_datasets = pd.concat([Legit_Transaction, Fraud_Sample], axis=0)

In [26]: new_datasets['Class'].value_counts()

Out[26]: 0    284315
         1    284315
         Name: Class, dtype: int64
```

Figure 8- New datasets containing equals number of legit and Fraud Transaction

e. Data Analysis

Data analysis involves the systematic examination and interpretation of data to uncover patterns, trends, relationships, and insights that address research questions or objectives. It plays a crucial role in transforming raw data into meaningful information, facilitating decision-making, hypothesis testing, and drawing conclusions[34].

a. Eventually, various statistical constraints of the datasets are identified to ensure proper analysis of the data. The table below illustrates the mean, count, standard deviation, minimum, and maximum amounts of transactions made for both legitimate and fraudulent transactions.

```
In [46]: Legit_Transaction.Amount.describe()

Out[46]: count    284315.000000
         mean      12026.313506
         std        6929.500715
         min          50.120000
         25%        6034.540000
         50%       11996.900000
         75%       18040.265000
         max       24039.930000
         Name: Amount, dtype: float64

In [47]: Fraud_Transaction.Amount.describe()

Out[47]: count    284315.000000
         mean      12057.601763
         std        6909.750891
         min          50.010000
         25%        6074.640000
         50%       12062.450000
         75%       18033.780000
         max       24039.930000
         Name: Amount, dtype: float64
```

Figure 9- Statistical Calculation base on Amount of Transaction

c. The dataset further analyzed by grouping data using credit_dt.groupby('Class').mean(). This is to classify data as fraud (Class 1) versus those that are not (Class 0).

credit_dt.groupby('Class'): This part groups the data by the 'Class' column, which typically contains binary values indicating whether a transaction is fraudulent or not. So, this groups the data into two groups: one for transactions classified as fraudulent (Class 1) and the other for legitimate transactions (Class 0).

.mean(): After grouping the data, .mean() calculates the mean value for each numerical column within each group. By doing so, it gives you the average values of various features (such as transaction amount, time of transaction, etc.) for both fraudulent and non-fraudulent transactions separately.

This allows us to compare the average values of different features between fraudulent and non-fraudulent transactions. Discrepancies in these averages can sometimes highlight patterns or characteristics that are indicative of fraudulent activity, which can then be used to build better fraud detection models.

```
In [19]: credit_dt.groupby('Class').mean()
```

Out[19]:

Class	id	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	
0	142442.987714	0.505761	-0.491878	0.682095	-0.736981	0.338839	0.435088	0.491234	-0.144294	0.585522	...	-0.179851	-0.10964	-0.014098	-0.010
1	426186.012286	-0.505761	0.491878	-0.682095	0.736981	-0.338839	-0.435088	-0.491234	0.144294	-0.585522	...	0.179851	0.10964	0.014098	0.010

2 rows × 30 columns

Figure 10- Mean values of Each Transaction

f. To prepare the dataset for logistic regression algorithms, the parameters are divided into two variables. Variable 'X' contains all the table schemas except for the 'class' attribute, which distinguishes between legitimate and fraudulent transactions. Meanwhile, variable 'Y' exclusively stores the 'class' schema, representing '0' for legitimate transactions and '1' for fraudulent ones. This step is essential as a prerequisite for training the dataset in logistic regression algorithms.

```
In [24]: X = new_datasets.drop(columns='Class', axis = 1)
         Y = new_datasets['Class']
```

Figure11–Variables stores status of transactions and other table schema

d. Modeling

Modeling using logistic regression involves using the logistic regression algorithm to build a predictive model that can classify transactions as either fraudulent or legitimate based on various features or attributes associated with each transaction[16].

The modeling begins by separating the overall data into training and test data. Below is the overview of the training and the test data;

i. Training Dataset:

The training dataset is a subset of the entire dataset that is used to train the logistic regression model. It consists of historical transaction data, where each transaction is labeled as either fraudulent or legitimate. This dataset is used by the model during the training process to learn the relationship between the input features (e.g., transaction amount, time of transaction, etc.) and the target variable (fraudulent or legitimate)[35].

ii. Test Dataset:

The test dataset is another subset of the entire dataset that is kept separate from the training dataset. It is used to evaluate the performance of the trained logistic regression model. The test dataset also consists of labeled transaction data, but the model has never seen this data during the training process. By evaluating the model's performance on unseen data, we can assess its ability to generalize to new, unseen transactions.

iii. How the training and test datasets works;

Training Phase: During the training phase, the logistic regression model is trained using only the training dataset. The model learns the patterns and relationships in the training data, adjusting its parameters to minimize the prediction error[35].

Evaluation Phase: After training, the model's performance is evaluated using the test dataset. The model makes predictions on the transactions in the test dataset, and these predictions are compared to the true labels (i.e., whether each transaction is fraudulent or legitimate). Evaluation metrics such as accuracy is calculated based on these predictions to assess the model's performance.

By using separate training and test datasets, we can obtain an unbiased estimate of the model's performance on new, unseen data. This helps to ensure that the model is not over fitting to the training data and that it generalizes well to real-world transactions.

The screenshot below illustrates how the overall data is divided into four variables for both the training and test sets. Additionally, the number of data points under each variable is highlighted in the screenshot.

```
In [98]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

In [99]: print (X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
         (454904, 30) (113726, 30) (454904,) (113726,)
```

Figure 12- splitting dataset; Training and Test

iv. Employing Logistic Regression Algorithm.

Currently, after completing preprocessing steps to clean and organize the data, it is prepared for analysis using logistic regression. In the line of code 'model = LogisticRegression()', we initialize an instance of the logistic regression algorithm, which sets the stage for building a predictive model. This line essentially creates a container named 'model' that holds all the necessary functions and properties of the logistic regression algorithm[16].

Following this initialization, the subsequent line of code from the figure below signifies the beginning of the training process. Training involves feeding our prepared datasets into the logistic regression algorithm. During this training phase, the algorithm scrutinizes the provided data, examining the features and patterns within each transaction. By adjusting various parameters, such as weights and biases, the algorithm iteratively learns from the dataset, gradually improving its understanding of the relationships between input features and the outcome we're trying to predict.

Through this iterative learning process, the algorithm constructs a model that encapsulates the learned relationships between the input variables (features) and the output variable (target). This model serves as a representation of how the algorithm perceives the underlying structure of the data. Ultimately, the goal is to develop a model that accurately predicts the outcome of future transactions based on their features, leveraging the insights gained during the training phase.

```
In [100]: #model training logisted regression
          model = LogisticRegression()

In [101]: model.fit(X_train, Y_train)
Out[101]: LogisticRegression()
```

Figure 13- Employing logistic Regression Algorithm

v. Evaluation of the Model

In this research, we assess the performance of our developed model through two crucial metrics: accuracy on training data and accuracy on test data[29].

Firstly, we measure the accuracy on the training data by employing the following process: the model is tasked with predicting the outcomes of the X_train dataset, which comprises the input variables used during the training phase. Subsequently, we calculate the accuracy score by comparing these

predicted values against the actual outcomes present in Y_train, which encapsulates the corresponding labels or target values for the training set. The resulting accuracy score ranges between 0 and 1, with higher scores indicating a more precise alignment of the model with the training data. Notably, we achieve an accuracy score of approximately 0.999, signifying a high level of success in training the model with the provided data.

Similarly, we evaluate the model's performance on unseen data through the accuracy on the test data. This evaluation entails soliciting predictions from the model for the test data, which it hasn't encountered previously. Remarkably, the model achieves a score of around 0.999 on the test data as well, mirroring the accuracy achieved on the training data. This parity in scores underscores the consistency and reliability of the model's performance. Essentially, the similarity in accuracy scores between the training and test data suggests that the model generalizes well beyond the data it was trained on, exhibiting robust predictive capabilities.

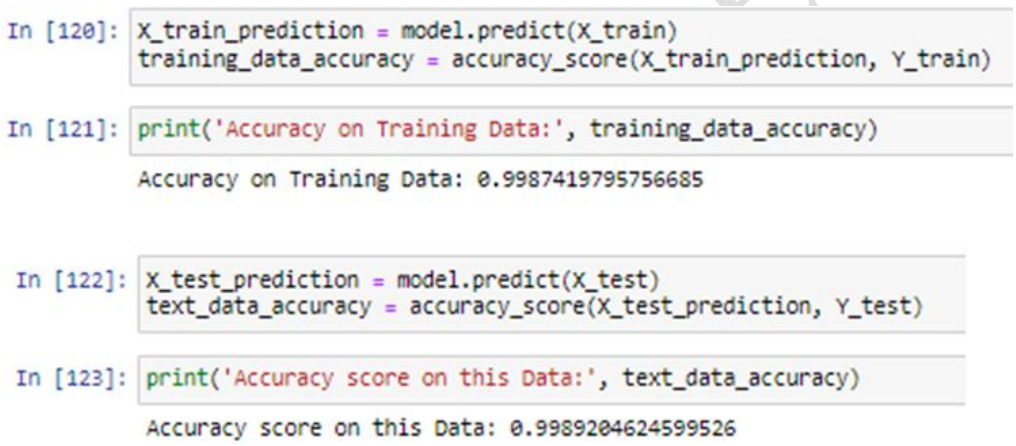

Figure 14- Evaluation

V. Results and Discussion

Logistic regression model was trained and evaluated using a dataset comprising credit card transactions with the aim of detecting fraudulent activities. The model's accuracy was meticulously assessed on both the training and test datasets to ascertain its performance.

On the training data, my model attained an impressive accuracy of 0.99, indicating its adeptness in learning from the provided examples. This high accuracy underscores the model's proficiency in understanding the intricacies of fraudulent and legitimate transactions within the training set.

Subsequently, on the test data, the model exhibited a similar accuracy of 0.99. This remarkable consistency between the training and test accuracies signifies the model's robustness in generalizing well to unseen data, a critical aspect in real-world applications.

The notable congruence in high accuracies attained on both training and test datasets indicates the logistic regression model's efficacy in accurately discerning between fraudulent and legitimate transactions. This performance sharply contrasts with the lower accuracy reported by[3], who achieved an accuracy of 0.94 using the same algorithm.

To provide a concise visual representation of the model's performance, a bar chart was meticulously crafted to illustrate its accuracy on both the training and test datasets. This graphical depiction serves as a compelling visualization of the model's consistent performance across diverse datasets, reinforcing the credibility of its efficacy in fraud detection.

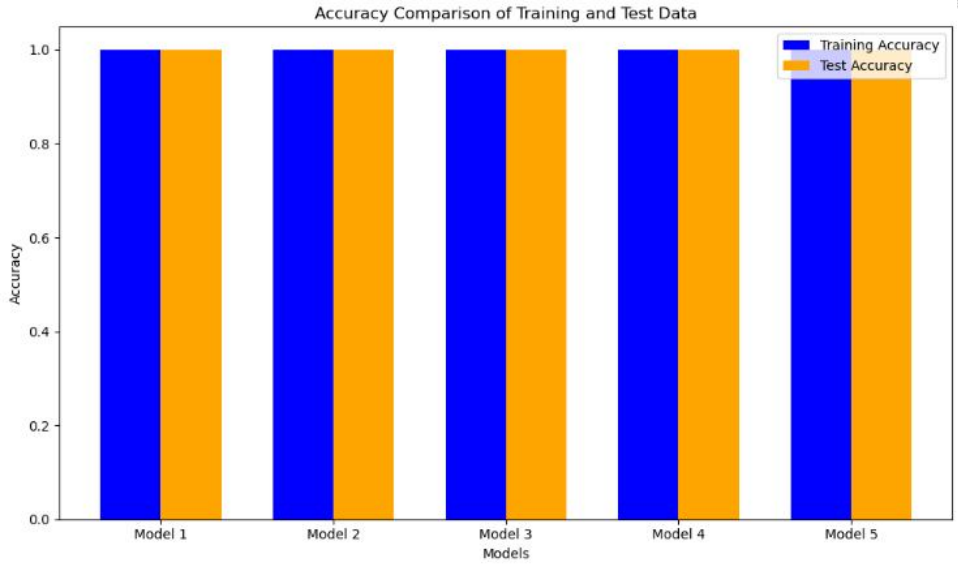

Figure 15- Accuracy in Test and Training

Transaction Status Prediction

In a concerted effort to comprehensively evaluate the efficacy of the logistic regression model, we conducted an extensive assessment of its predictive capabilities on unseen transactions. This evaluation aimed to ascertain the model's proficiency in accurately discerning the status—whether fraudulent or legitimate—of transactions previously unseen during the training phase.

The logistic regression model was deployed to analyze a meticulously curated dataset comprising transactions that had not been encountered during the model's training phase. Subsequently, the predicted statuses generated by the model were compared against the ground truth labels to gauge the model's predictive accuracy.

The findings of this evaluation reveal an impressive performance by the logistic regression model, with an accuracy rate of 99.89% in predicting the status of unseen transactions. Specifically, the model correctly identified over 55,000 transactions as legitimate, demonstrating its adeptness in

discerning genuine transactions. Moreover, the model accurately flagged approximately 5,000 transactions as fraudulent, underscoring its efficacy in identifying potentially illicit activities within the dataset.

To provide a visual representation of the distribution of predicted transaction statuses, a meticulously crafted bar chart was generated. This graphical depiction elucidates the model's predictive prowess by illustrating the distribution of predicted statuses—fraudulent or legitimate—across the unseen dataset.

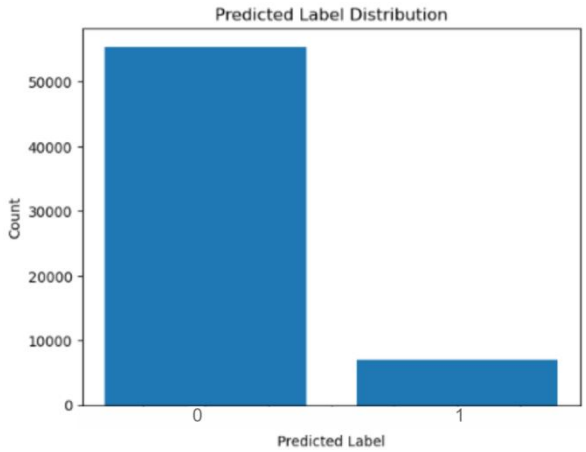

Figure 16- Result on Legit and Fraudulent Transactions

The obtained results demonstrate the effectiveness of the logistic regression model in detecting credit card fraud transactions. The high accuracies achieved on both the training and test datasets suggest that the model has successfully learned meaningful patterns from the data and can generalize well to unseen transactions.

However, it's important to note that the model's performance may vary depending on the characteristics of the dataset and the features used for training. Further analysis is warranted to identify potential areas for improvement and to assess the robustness of the model across different scenarios.

VI. Conclusion

In conclusion, the logistic regression model stands as a formidable asset in the continuous endeavor to combat financial fraud. Through its capacity to detect subtle irregularities within credit card transactions and its impressive ability to generalize to unseen data, the model emerges as a cornerstone in the arsenal of fraud detection mechanisms. These findings not only highlight the efficacy of advanced analytical techniques but also underscore the critical importance of fortifying financial ecosystems to preserve the integrity of transactions in an increasingly digitized landscape.

The study's revelations regarding the logistic regression model's proficiency in distinguishing between fraudulent and legitimate transactions align with broader trends in the field of fraud detection. By leveraging sophisticated algorithms and comprehensive datasets, researchers and practitioners alike can enhance their ability to detect and prevent fraudulent activities, thereby safeguarding financial systems and bolstering consumer trust.

Moreover, the model's robust generalization capabilities signify its adaptability to evolving fraud patterns and emerging threats. This adaptability is paramount in an environment characterized by rapid technological advancements and increasingly sophisticated fraudulent schemes.

Looking ahead, further research and development efforts are warranted to continuously refine and improve fraud detection methodologies. Collaborative endeavors between academia, industry, and regulatory bodies can foster innovation and drive the adoption of cutting-edge technologies in the fight against financial fraud.

Ultimately, the findings of this study underscore the pivotal role of the logistic regression model and advanced analytical techniques in fortifying financial ecosystems. By embracing these tools and leveraging data-driven insights, stakeholders can work towards a future where financial transactions are conducted with heightened security and confidence, safeguarding both individual consumers and the broader economy against the pervasive threat of fraud.

Reference

[1] S. Kiruthika and C. N. Sowmyarani, “Credit Card Fraud Detection using Machine Learning and Deployment of Model in Public Cloud as a Web Service,” vol. 3878, no. 2, pp. 548–552, 2020, doi: 10.35940/ijrte.B3800.079220.

[2] K. B. Soni, M. Chopade, and R. Vaghela, “Credit Card Fraud Detection Using Machine Learning Approach,” vol. 4, no. 2, pp. 71–76, 2021.

[3] V. K. K. S, K. V. G. Vijaya, V. S. A, and K. Pratibha, “Credit Card Fraud Detection using Machine Learning Algorithms,” vol. 9, no. 07, pp. 1526–1530, 2020.

[4] N. Purohit and R. G. Vishwakarma, “Credit Card Fraud Detection Using Machine Learning Algorithms Using Python Technology,” vol. 18, no. 6, pp. 7995–8006, 2021.

[5] A. Alali and A. Alali, “Financial Fraud Detection using Machine Learning Techniques,” 2020.

[6] C. Journal, “COOU Journal of Physical S ciences 3(1), 2020,” vol. 3, no. 1, pp. 493–498, 2020.

[7] R. Dekou *et al.*, “Machine Learning Methods for Detecting Fraud in Online Marketplaces .,” no. July, pp. 1–8, 2021.

[8] B. P. Swarna and J. A. P. S. S, “Credit card fraud detection system using machine learning,” vol. 8, no. 7, pp. 249–252, 2021.

[9] S. Kumar and H. Kumar, “Credit Card Fraud Detection Using Machine Learning,” vol. 11, no. 6, pp. 60–67, 2021.

[10] G. Kashyap and U. Raj, “Credit Card Fraud Detection using machine learning SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY , GREATER NOIDA,” 2021.

[11] S. Han, “Credit Card Fraud Detection With Machine Learning,” 2020.

[12] K. S. Lim, L. H. Lee, and Y. Sim, “A Review of Machine Learning Algorithms for Fraud Detection in Credit Card Transaction,” vol. 21, no. 9, 2021.

[13] T. Kabir, T. Nishat, and S. B. Tory, “Credit Card Fraud Detection Using Machine Learning Techniques,” no. September, 2021.

[14] P. Sharma, S. Banerjee, D. Tiwari, and J. C. Patni, “Machine Learning Model for Credit Card Fraud Detection- A Comparative Analysis,” vol. 18, no. 6, pp. 789–796, 2021.

[15] P. S. Krishna, “CREDIT CARD FRAUD DETECTION USING LOGISTIC REGRESSION,” vol. 11, no. 4, pp. 471–477, 2020.

[16] M. Alemad, “Credit Card Fraud Detection Using Machine Learning,” 2022.

[17] H. Cheng, “Credit Card Fraud Detection Using Logistic Regression and Machine Learning Algorithms,” vol. 2, p. 44, 2023, [Online]. Available: https://escholarship.org/uc/item/18d8w42r.

[18] K. Dian, “SCHOOL OF SCIENCE AND ENGINEERING CREDIT CARD FRAUD DETECTION USING DEEP,” 2022.

[19] M. Dawaki, “Fraudulent Text Detection System Using Hybrid Machine Learning and Natural Language Processing Approaches,” vol. 7, no. 10, pp. 1110–1118, 2022.

[20] S. Patil, P. Thakre, G. Tupe, and P. Kenjale, “International Journal of Research Publication and Reviews Journal homepage : www.ijrpr.com ISSN 2582-7421 CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING AND DATA SCIENCE,” vol. 3, no. 5, pp. 3728–3735, 2022.

[21] R. R. Devi, “Credit Card Fraud Detection Using AI / ML / CNN,” vol. 6, no. 9, pp. 242–249, 2023.

[22] K. Nikhil, B. V. Maharshi, and K. Tanooj, “Credit card fraud detection using machine learning algorithms,” vol. 14, no. 04, pp. 471–485, 2023.

[23] D. Shah and L. K. Sharma, “Credit Card Fraud Detection using Decision Tree and Random Forest,” vol. 02012, 2023.

[24] G. Sandhya, M. Abishek, S. G. Kumar, and R. S. J. Kumar, “Credit Card Fraud Detection using Machine Learning Algorithms Credit Card Fraud Detection using,” no. January, 2023, doi: 10.1007/978-981-19-5221-0.

[25] P. P. Sammelwerksbeitrag, “Scraping social media data as platform research : A data hermeneutical perspective,” 2023, [Online]. Available: www.ssoar.info Scraping.

[26] M. G. S. Eswari, M. A. S. Malleswari, M. S. Sakina, M. K. Anjali, M. D. N. S. Sindhu, and M. J. R. Visha, “International Journal of Engineering Technology Research & Management International Journal of Engineering Technology Research & Management,” no. 02, pp. 9–12, 2022.

[27] E. A. A. Abuhamda, I. A. Ismail, and T. English, “Understanding quantitative and qualitative

research methods : A theoretical perspective for young researchers Understanding Quantitativeand Qualitative ResearchMethods : A TheoreticalPerspective for Young Researchers," no. February, pp. 70–87, 2021, doi: 10.2501/ijmr-201-5-070.

[28] T. Hayashi, T. Shimizu, and Y. Fukami, "Collaborative Problem Solving on a Data Platform," vol. 120, no. 362, pp. 37–40, 2021.

[29] B. Chowdari and S. Parthiban, "International Journal of Research Publication and Reviews Credit Card Fraud Detection using Logistic Regression Compared with t-SNE to Improve Accuracy," vol. 3, no. 8, pp. 1000–1004, 2022.

[30] O. Scrivner, "Introduction to Web Scraping Using Python," p. 2018.

[31] A. Ramaswamy, A. D. A. Mulimani, S. Pal, A. K. Singh, and H. G. Rani, "CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING AND DEEP LEARNING," no. June, pp. 775–777, 2021.

[32] H. Z. Alenzi and N. O. Aljehane, "Fraud Detection in Credit Cards using Logistic Regression," vol. 11, no. 12, pp. 540–551, 2020.

[33] V. Agarwal, "Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis," no. December 2015, 2016, doi: 10.5120/ijca2015907309.

[34] J. V. N. Lakshmi, "Machine learning techniques using python for data analysis in performance evaluation Machine learning techniques using python for data analysis in performance evaluation," no. January 2018, 2019, doi: 10.1504/IJISTA.2018.10012853.

[35] V. Tech and A. Virginia, "Systematic Training and Testing for Machine Learning Using Combinatorial Interaction Testing a National," 2022.