Final Year Dissertation

On

Topic: fraud detection on online transaction using machine learning algorithm

Submitted to

In Partial Fulfilment of the requirements for the degree of Bachelor of Science and Information Technology (BscIT)

Submitted by

Group no:

En. no. : 202101619010014	Name : Ansh Dankhara
En. no. : 202101619010017	Name : Kenil Dhola
En. no. : 202101619010025	Name : Fenil Kakadiya

Batch 2021-24

Under the guidance of

Prof. Tripti Dodiya

GLS UNIVERSITY

Faculty of Computer Applications And Information Technolody

*I*MSc(*IT*) Programme

CERTIFICATE

This is to verify and certify that

1) Ansh Dankhara
2) Kenil Dhola
3) Fenil Kakadiya

Students of Sem-VI , Integrated MscIT , TY iMScIT,

FCAIT , GLS University has/have Successfully completed the

Dissertation

On

"fraud detection on online transaction using machine learning algoridham"

As a partial fulfillment of the study of Third Year Sem-VI ,IMscIT

Date of submission : 07.04.2024

Prof. Tripti Dodiya
(Project Guide)

Prof. Tripti Dodiya
(Project Co-ordinator)

Acknowledgements

I honor my mentor, Prof. Nasrin , for the steadfast support with the initial investigation with immense appreciation and gratefulness. Your guidance will never be forgotten!

To Prof. Tripti Dodiya, who have joined us from day 1 till now and making this quantitative-inclined researcher embracing his qualitative methods with enthusiasm. Your Interest and Support have been greatly appreciated . It has been the highest honor to have you as our guide; I thank you!

Abstract :

The rapid proliferation of online transactions in the contemporary digital landscape has accentuated the critical need for robust mechanisms to detect and prevent fraudulent activities. As financial interactions increasingly migrate to digital platforms, the threat landscape for fraudulent endeavors becomes more sophisticated and elusive.

This documentation endeavors to address this pressing challenge by exploring the efficacy of machine learning algorithms in fortifying online transactions against fraudulent activities. The primary objective of this study is to elucidate the application of state-of-the-art machine learning techniques in the realm of fraud detection. Through a meticulous exploration of diverse algorithms, encompassing supervised and unsupervised learning paradigms, this research aims to unveil the inherent capabilities and limitations of these methodologies.

The document offers a nuanced understanding of the machine learning landscape, delving into the intricacies of feature engineering, model training, and performance evaluation specific to fraud detection contexts. In the pursuit of knowledge and technological advancements, this documentation synthesizes existing literature, critically appraises established methodologies, and proposes innovative approaches to enhance the efficacy of fraud detection systems.

The research methodology involves the curation of a comprehensive dataset representative of real-world online transactions, ensuring the applicability of findings to practical scenarios. The results and analyses presented herein provide a detailed examination of the performance metrics of various machine learning models in detecting fraudulent transactions. The discussion section interprets these findings in the broader context of online fraud detection, emphasizing practical implications and potential avenues for future research.

Ethical considerations inherent in handling sensitive transactional data are given meticulous attention, reflecting a commitment to user privacy and data integrity. this documentation contributes not to the academic discourse surrounding fraud detection and machine learning but also presents practical insights for industry practitioners, policymakers, and stakeholders invested in securing digital financial ecosystems. It is anticipated that the findings and methodologies encapsulated herein will inform and inspire further advancements in the perpetual battle against online fraud.

CHAPTER - I : Introduction

In the dynamic landscape of modern commerce, the proliferation of online transactions has ushered in unprecedented convenience but concurrently unveiled a burgeoning threat – fraudulent activities that exploit the vulnerabilities of digital financial ecosystems. As financial interactions traverse the digital realm, the paradigm for perpetrating fraud evolves with unparalleled sophistication, necessitating the deployment of innovative and adaptive countermeasures.

This documentation embarks on an exploratory journey into the intricate domain of "Fraud Detection in Online Transactions using Machine Learning Algorithms." The imperative to safeguard digital transactions against fraudulent activities has become a paramount concern for individuals, businesses, and financial institutions alike. The conventional methods of fraud detection, while stalwart in their intent, grapple with the escalating complexities of cyber threats.

This study addresses this burgeoning challenge by delving into the transformative potential of machine learning algorithms, a frontier of technological prowess that has exhibited remarkable efficacy in deciphering patterns, discerning anomalies, and fortifying digital landscapes against illicit activities.

Our research agenda is grounded in the pursuit of a holistic understanding of the intricate interplay between online transactions and machine learning algorithms.. The study is underpinned by a commitment to methodological rigor, employing diverse machine learning techniques in a quest to discern fraudulent patterns within the vast expanse of transactional data. As we navigate the research landscape, ethical considerations guide our every step. Privacy, transparency, and responsible data handling are integral tenets that underlie the ethical framework of this study.

The utilization of real-world transactional datasets underscores our commitment to practical relevance, ensuring that the findings resonate with the challenges faced by contemporary digital financial systems. This documentation does not merely aspire to contribute to the academic discourse; it endeavors to furnish actionable insights for industry stakeholders, policymakers, and cybersecurity practitioners. The ensuing chapters will unfold the intricacies of our methodology, present empirical findings, and chart a course towards a future where the nexus of technology and security fortifies the integrity of online financial ecosystems.

In the transformative landscape of contemporary commerce, the exponential rise of online transactions heralds a new era of connectivity, convenience, and accessibility. The digital paradigm, woven into the fabric of everyday life, propels financial interactions beyond the confines of brick-and-mortar institutions into the dynamic realm of cyberspace. The genesis of digital transactions traces its roots to the nascent stages of the internet. The ubiquity of smartphones and the democratization of high-speed internet have propelled this evolution, fundamentally reshaping the way individuals and businesses engage in financial transactions.

Concomitant with the surge in online transactions, a parallel narrative unfolds – the ascendancy of fraudulent activities seeking to exploit the vulnerabilities inherent in digital financial landscapes. The ingenuity of cybercriminals, evolving from simplistic scams to sophisticated tactics such as identity theft and intricate fraud networks, poses an escalating threat. The historical underpinnings of fraud, intertwined with the growth of digital transactions, underscore the imperative to fortify these ecosystems against the persistent and adaptive nature of illicit endeavors.

Traditional methodologies, rooted in rule-based systems and static heuristics, once stalwarts in the fight against fraud, find themselves grappling with the dynamic and complex nature of contemporary cyber threats. The limitations of predefined rules become glaringly evident as fraud patterns metamorphose in response to evolving technologies and strategies employed by malicious actors. The need for innovative, adaptive, and data-driven approaches becomes imperative to stay ahead in the perpetual cat-and-mouse game between defenders and perpetrators.

As we embark on this exploratory journey into the realm of "Fraud Detection in Online Transactions using Machine Learning Algorithms," the roadmap ahead unfolds with the promise of not only advancing academic discourse but also imparting actionable insights for industry practitioners, policymakers, and stakeholders invested in the resilience of digital financial ecosystems. The subsequent chapters will unravel the intricacies of the research methodology, present empirical findings, and chart a course towards a future where technology, ethics, and innovation converge to fortify the integrity of online financial interactions.

CHAPTER - II : BACKGROUND

The advent of the digital age has witnessed a monumental shift in the way financial transactions are conducted, with a significant portion transitioning to online platforms. This paradigm shift, while ushering in unprecedented convenience and accessibility, has concurrently given rise to a pressing challenge – the escalating threat of fraudulent activities targeting digital financial ecosystems. Understanding the background of this challenge necessitates a retrospective exploration into the evolution of online transactions and the historical trajectory of fraud detection methodologies.

Evolution of Online Transactions:

The journey of online transactions dates back to the nascent stages of the internet, where rudimentary forms of digital commerce began to take shape. As technology advanced, online transactions evolved from basic electronic payments to intricate financial interactions, encompassing e-commerce, online banking, and a plethora of digital services. The ubiquity of smartphones and the widespread availability of high-speed internet further catalyzed the surge in online transactions, transforming the global economic landscape.

Rise of Fraudulent Activities:

Concomitant with the surge in online transactions, fraudulent activities assumed new dimensions, exploiting the vulnerabilities inherent in digital financial ecosystems. The methods employed by fraudsters evolved from simple scams to sophisticated techniques such as identity theft, account takeover, and the creation of intricate fraudulent networks. Traditional methods of fraud detection, reliant on rule-based systems and static heuristics, found themselves increasingly inadequate in the face of dynamic and complex cyber threats.

Limitations of Traditional Approaches:

The inadequacies of traditional fraud detection methods became apparent as the scale and sophistication of fraudulent activities outpaced the capabilities of rule-based systems. leading to high false positive rates and a delayed response to emerging threats. Recognizing the limitations of these conventional approaches became imperative for the financial industry to stay ahead in the perpetual cat-and-mouse game with cybercriminals.

Emergence of Machine Learning in Fraud Detection:

In response to the evolving landscape of online fraud, the emergence of machine learning marked a paradigm shift in fraud detection methodologies. Machine learning algorithms, capable of learning patterns from data and adapting to dynamic scenarios, presented a potent solution to the shortcomings of traditional approaches. These algorithms could analyze vast datasets, identify subtle patterns indicative of fraudulent behavior, and continuously evolve to counter new and sophisticated fraud tactics.

Understanding the background of fraud detection in online transactions, therefore, entails recognizing the historical context of the digital revolution, the parallel rise in fraudulent activities, the limitations of traditional approaches, and the transformative impact of machine learning on fortifying the resilience of digital financial ecosystems. This backdrop sets the stage for a comprehensive exploration into the methodologies, challenges, and innovations that define the contemporary landscape of fraud detection in online transactions.

CHAPTER - III : LITERATURE REVIEW

A comprehensive literature review is fundamental to contextualize the current study within the broader landscape of fraud detection in online transactions using machine learning algorithms. This synthesis of existing research offers insights into established methodologies, identifies gaps in knowledge, and paves the way for innovative contributions to the field.

Foundations of Fraud Detection:

Early literature on fraud detection lays the foundations for understanding the historical evolution of methods employed in identifying and preventing fraudulent activities. Rule-based systems and heuristic approaches dominated initial research, showcasing attempts to create predefined rules that could capture known patterns of fraud.

Transition to Machine Learning:

As the limitations of rule-based systems became evident, the literature reflects a shift towards machine learning as a potent solution. Seminal studies explore the application of various machine learning algorithms, from classical approaches like decision trees and logistic regression to more advanced methods such as support vector machines and ensemble methods.

Supervised and Unsupervised Learning:

Literature in this domain delves into the comparative analysis of supervised and unsupervised learning techniques. Supervised learning, where models are trained on labeled datasets, proves effective in detecting known patterns of fraud. Unsupervised learning, on the other hand, excels in identifying novel and emerging fraud patterns without predefined labels.

Feature Engineering and Selection:

Studies on feature engineering elucidate the significance of selecting relevant variables for effective fraud detection. Researchers explore the nuances of creating meaningful features that encapsulate. This involves understanding the temporal, spatial, and behavioral aspects of transactions that are indicative of potential fraud.

Challenges in Imbalanced Datasets:

The literature review also unveils the challenges posed by imbalanced datasets, where instances of fraud are significantly outnumbered by legitimate transactions. Techniques such as oversampling, undersampling, and the use of cost-sensitive learning algorithms are scrutinized in addressing this imbalance and mitigating the impact on model performance.

Ensemble Methods and Hybrid Approaches:

Emerging trends in the literature showcase a surge in interest in ensemble methods and hybrid approaches. Ensemble methods, amalgamating multiple models to enhance predictive power, are explored for their effectiveness in dealing with complex fraud patterns. Hybrid models, combining rule-based systems with machine learning algorithms, aim to leverage the strengths of both approaches.

Explanability and Interpretability:

Recent literature emphasizes the importance of model explainability and interpretability. As machine learning models become more complex, understanding how they reach decisions becomes critical for user trust and regulatory compliance. Studies investigate techniques to make machine learning models more transparent and interpretable.

Cross-Industry Insights:

The literature review transcends the confines of a singular industry, offering cross-industry insights into fraud detection. Studies explore applications of machine learning in diverse sectors, from finance and healthcare to e-commerce and telecommunications, elucidating the transferability of methodologies and the contextual nuances that influence model effectiveness.

Gaps and Future Directions:

While existing literature contributes significantly to the understanding of fraud detection in online transactions, identified gaps in knowledge pave the way for the current study. These gaps may include unexplored algorithmic frameworks, specific challenges in emerging fraud patterns, or the need for enhanced explainability in machine learning models.

Historical Perspectives on Machine Learning:

Examining the historical trajectory of machine learning in fraud detection provides valuable insights. Early studies focused on conventional algorithms, showcasing the effectiveness of decision trees and logistic regression. The literature illustrates a paradigm shift towards more sophisticated models, including support vector machines, random forests, and neural networks, as computational capabilities advanced.

Mitigation Strategies:

The challenges posed by imbalanced datasets, where instances of fraud are sparse compared to legitimate transactions, emerge as a focal point. The literature scrutinizes techniques such as oversampling, undersampling, and the utilization of cost-sensitive learning algorithms to mitigate the impact of class imbalance, ensuring models maintain efficacy in the face of skewed data distributions.

Ensemble Methods and Hybrid Approaches:

Recent literature illuminates the prominence of ensemble methods and hybrid approaches. Ensemble models, combining the strengths of multiple algorithms, showcase heightened predictive power. Hybrid approaches, integrating rule-based systems with machine learning, provide a nuanced perspective, leveraging both interpretability and adaptability in fraud detection.

Explainability and Interpretability Frameworks:

The growing emphasis on model explainability and interpretability emerges as a critical trend. Researchers delve into frameworks such as LIME and SHAP to demystify the decision-making processes of complex machine learning models. Enhancing the transparency of models becomes imperative for user trust and regulatory compliance.

Cross-Industry Applications and Lessons:

Extending beyond the financial sector, the literature explores cross-industry applications of machine learning in fraud detection. Insights from healthcare, e-commerce, and telecommunications reveal transferable methodologies and contextual nuances that enrich the overall understanding of fraud detection in diverse domains.

Global Perspectives and Regulatory Considerations:

An intriguing facet within the literature review involves global perspectives on fraud detection. Studies analyze regulatory frameworks, cultural variations, and regional nuances influencing fraud dynamics. A holistic understanding of the global landscape provides a comprehensive foundation for designing adaptable and culturally sensitive fraud detection systems.

Emerging Technologies and Future Directions:

A forward-looking trend within the literature review pertains to emerging technologies. Researchers contemplate the integration of blockchain, artificial intelligence, and edge computing in augmenting fraud detection capabilities. These explorations set the stage for future directions in the continual evolution of technology-driven fraud prevention.

In summation, the literature review serves as a scaffold, underpinning the current study within the rich tapestry of research on fraud detection in online transactions. The synthesis of diverse perspectives, methodologies, and challenges from existing literature forms the bedrock upon which the current research builds, contributing to the collective knowledge base of machine learning applications in fortifying digital financial ecosystems against fraudulent activities.

CHAPTER - IV : METHODOLOGY

What is SVM model?

A popular machine learning approach in credit card fraud detection models is called Support Vector Machine (SVM). A supervised learning technique that can be applied to tasks involving regression and classification is SVM. SVM is typically used for binary classification in the context of credit card fraud detection, where the objective is to discern between real (non-fraudulent) and fraudulent transactions.

Here's a brief overview of how SVM works in the context of credit card fraud detection:

1. Training Phase:

- Data Preparation: Historical credit card transaction data is collected and labeled as either genuine or fraudulent.
- Feature Extraction: Relevant features (such as transaction amount, location, time, etc.) are extracted from the data to create a feature matrix.
- Labeling: Each transaction is labeled as either a legitimate or fraudulent one.

2. Model Training:

- The SVM algorithm aims to find a hyperplane that best separates the two classes (genuine and fraudulent transactions) in the feature space. This hyperplane is chosen in such a way that it maximizes the margin between the two classes.
- SVM can use different kernel functions (linear, polynomial, radial basis function, etc.) to map the input data into a higher-dimensional space, making it easier to find a separating hyperplane.
- Here we use 80:20 ratio for training and testing.

3. Testing and Prediction:

- Once the SVM model is trained, it can be used to predict the likelihood of fraud for new, unseen transactions.
- The model assigns a predicted class (fraudulent or genuine) to each transaction based on its feature values.

4. Evaluation:

- The performance of the SVM model is evaluated using metrics such as precision, recall, accuracy, and F1-score.
- The model's ability to correctly identify fraudulent transactions while minimizing false positives is crucial in credit card fraud detection.

ALGORIDHM TESTING ON DATASET:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import seaborn as sns


data = pd.read_csv("/content/creditcard.csv")


print(data.isnull().sum())
data.isnull().fillna(0, inplace=True)
print(data.info)
```

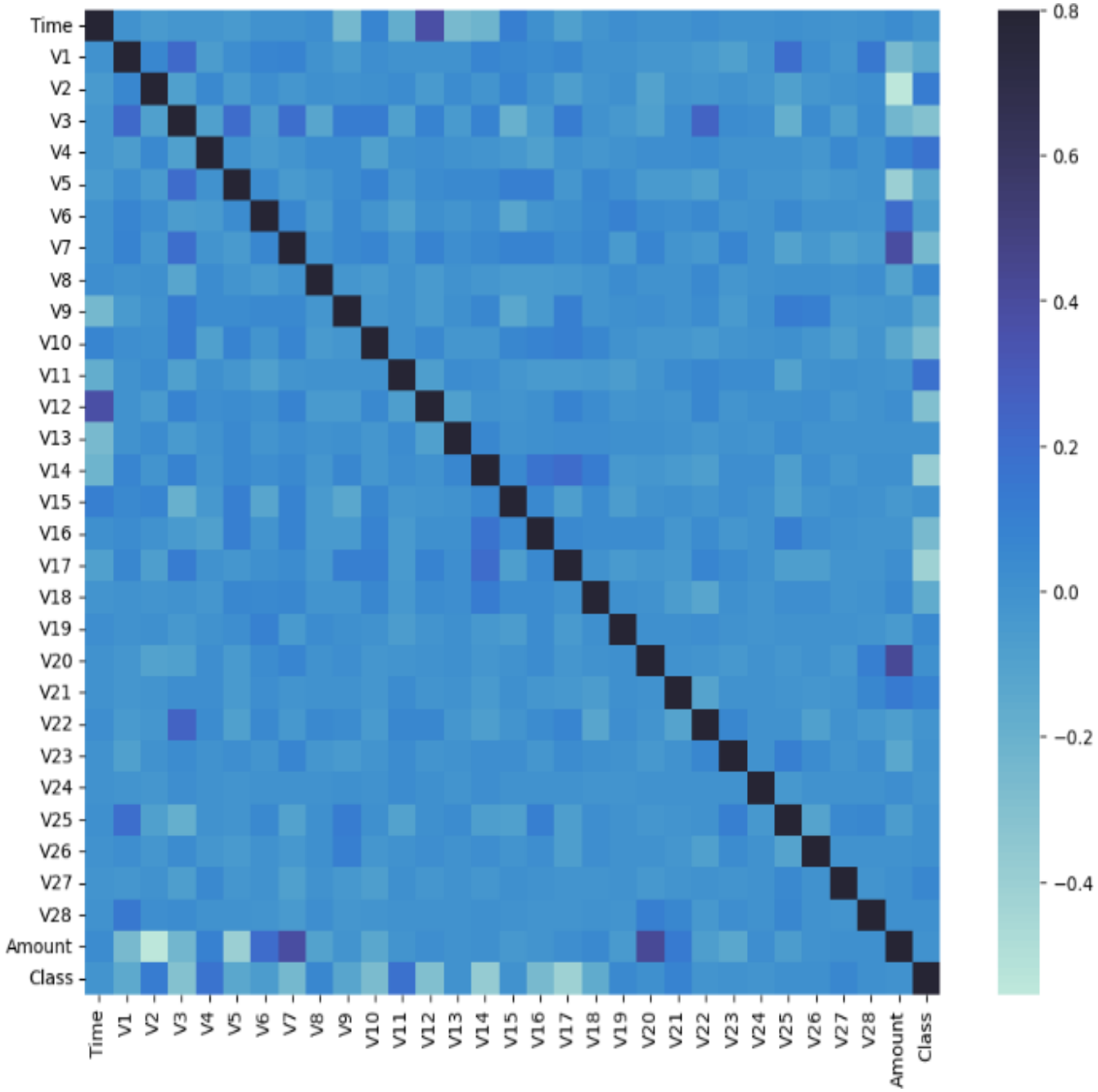
```
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       1
V25       1
V26       1
V27       1
V28       1
Amount    1
Class     1
```

```
<bound method DataFrame.info of          Time        V1        V2        V3        V4
0           0 -1.359807 -0.072781  2.536347  1.378155 -0.338321  0.462388
1           0  1.191857  0.266151  0.166480  0.448154  0.060018 -0.082361
2           1 -1.358354 -1.340163  1.773209  0.379780 -0.503198  1.800499
3           1 -0.966272 -0.185226  1.792993 -0.863291 -0.010309  1.247203
4           2 -1.158233  0.877737  1.548718  0.403034 -0.407193  0.095921
...       ...       ...       ...       ...       ...       ...       ...
105062  69348  1.255690 -1.393540  0.692738 -1.289430 -1.682706 -0.276840
105063  69349 -1.250199  0.123183  1.051709 -0.236427  1.611834  0.284241
105064  69349 -1.925153 -0.140133  2.456659  0.148112 -0.284164 -0.150520
105065  69349  1.277856 -0.027651  0.249167  0.697814 -0.232188 -0.118324
105066  69350 -1.176438  0.811442  0.539281 -1.505794  0.401363  0.202677

              V7        V8        V9  ...       V21       V22       V23  \
0       0.239599  0.098698  0.363787  ... -0.018307  0.277838 -0.110474
1      -0.078803  0.085102 -0.255425  ... -0.225775 -0.638672  0.101288
2       0.791461  0.247676 -1.514654  ...  0.247998  0.771679  0.909412
3       0.237609  0.377436 -1.387024  ... -0.108300  0.005274 -0.190321
4       0.592941 -0.270533  0.817739  ... -0.009431  0.798278 -0.137458
...          ...       ...       ...  ...       ...       ...       ...
105062 -1.136732 -0.076399 -1.738348  ...  0.039513  0.368717 -0.072065
105063  0.288190  0.303990 -0.430526  ...  0.034681 -0.039897 -0.049460
105064 -1.129736  1.003991 -0.218409  ...  0.447094  0.895815 -0.233737
105065 -0.145313 -0.048568  0.677505  ... -0.141950 -0.164073 -0.194695
105066  0.284898  0.547988 -0.438512  ...  0.077511  0.008094 -0.020608

             V24       V25       V26       V27       V28  Amount  Class
0       0.066928  0.128539 -0.189115  0.133558 -0.021053  149.62    0.0
1      -0.339846  0.167170  0.125895 -0.008983  0.014724    2.69    0.0
2      -0.689281 -0.327642 -0.139097 -0.055353 -0.059752  378.66    0.0
3      -1.175575  0.647376 -0.221929  0.062723  0.061458  123.50    0.0
4       0.141267 -0.206010  0.502292  0.219422  0.215153   69.99    0.0
...          ...       ...       ...       ...       ...     ...    ...
105062  0.106986  0.257991 -0.087901  0.054217  0.043817  116.00    0.0
105063 -1.399614  0.217418 -0.496523 -0.042176  0.161224    3.99    0.0
105064  0.262490  0.050375  0.515140  0.207773 -0.090777    4.69    0.0
105065 -0.381716  0.696330  0.489582 -0.017759  0.002991   10.00    0.0
105066       NaN       NaN       NaN       NaN       NaN     NaN    NaN

[105067 rows x 31 columns]>
```

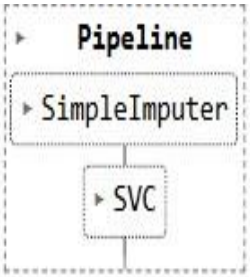
```
X = data.iloc[:, 0:30]
y = data.Class
Fraud = data[data['Class'] == 1]
print(Fraud.Amount.describe())
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))
```

```
count     232.000000
mean      115.520474
std       253.721393
min         0.000000
25%         1.000000
50%         7.595000
75%        99.990000
max      1809.680000
Name: Amount, dtype: float64
Fraud Cases: 232
Valid Transactions: 104834
```

```
corr = data.corr()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corr, vmax = .8, square = True,center=1)
plt.show()
```


Time
V1
V2
V3
V4
V5
V6
V7
V8
V9
V10
V11
V12
V13
V14
V15
V16
V17
V18
V19
V20
V21
V22
V23
V24
V25
V26
V27
V28
Amount
Class
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

```
X_train, Xtest, y_train, ytest = train_test_split(X, y, test_size=0.25)


from sklearn.impute import SimpleImputer
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


pipeline = make_pipeline(SimpleImputer(strategy='mean'), SVC())


pipeline.fit(X_train, y_train)
```




```
predictions = pipeline.predict(X_test)
```

```
X = X.dropna()
```

```
classifier = SVC()
classifier.fit(X_train, y_train)
```


```
y_pred = classifier.predict(Xtest)
print("Accuracy score for this model is :", accuracy_score(ytest, y_pred))
print("Precision score for this model is :", precision_score(ytest, y_pred,zero_division=True))
print("Recall score for this model is :", recall_score(ytest, y_pred,zero_division=True))
print("F1 score for this model is :", f1_score(ytest, y_pred,zero_division=True))
```

```
Accuracy score for this model is : 0.9977919061940839
Precision score for this model is : 1.0
Recall score for this model is : 0.0
F1 score for this model is : 0.0
```

What is Random Forest Algorithm?

Random Forest is an ensemble learning algorithm that can be used for credit card fraud detection. It belongs to the family of tree-based models and is known for its ability to handle complex relationships in data, reduce overfitting, and provide robust performance. Here's an overview of how the Random Forest algorithm works in the context of credit card fraud detection:

1. Data Collection and Preparation:

- Gather historical credit card transaction data with labeled cases of fraud and non-fraud.
- Extract relevant features such as transaction amount, location, time, etc.

2. Ensemble of Decision Trees:

- Random Forest builds an ensemble of decision trees during the training phase.
- Each decision tree is trained on a random subset of the data (sampling with replacement), and a random subset of features is considered at each split point.
- This randomness helps in decorrelating the individual trees, making the model more robust and less prone to overfitting.

3. Training:

- For each decision tree:
    - Randomly select a subset of the data (bootstrapping).
    - Randomly select a subset of features at each split point.
    - Grow the tree by recursively splitting the data based on the selected features until a stopping criterion is met (e.g., maximum depth reached, minimum samples per leaf).

4. Evaluation:

- Evaluate the Random Forest model using standard classification metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC-ROC).
- Random Forest can handle imbalanced datasets well, making it suitable for credit card fraud detection where the number of fraudulent transactions is often much smaller than legitimate ones.

ALGORIDHM TESTING ON DATASET:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


df = pd.read_csv("/content/creditcard.csv")


df.head()
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8
0	0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698
1	0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102
2	1	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676
3	1	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436
4	2	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533

5 rows × 31 columns

```
df.isnull().sum()
```

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       1
V19       1
V20       1
V21       1
V22       1
V23       1
V24       1
V25       1
V26       1
V27       1
V28       1
Amount    1
Class     1
dtype: int64
```

```
df.drop_duplicates()
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5974 entries, 0 to 5973
Data columns (total 31 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Time    5974 non-null   int64  
 1   V1      5974 non-null   float64
 2   V2      5974 non-null   float64
 3   V3      5974 non-null   float64
 4   V4      5974 non-null   float64
 5   V5      5974 non-null   float64
 6   V6      5974 non-null   float64
 7   V7      5974 non-null   float64
 8   V8      5974 non-null   float64
 9   V9      5974 non-null   float64
 10  V10     5974 non-null   float64
 11  V11     5974 non-null   float64
 12  V12     5974 non-null   float64
 13  V13     5974 non-null   float64
 14  V14     5974 non-null   float64
 15  V15     5974 non-null   float64
 16  V16     5974 non-null   float64
 17  V17     5974 non-null   float64
 18  V18     5973 non-null   float64
 19  V19     5973 non-null   float64
 20  V20     5973 non-null   float64
 21  V21     5973 non-null   float64
 22  V22     5973 non-null   float64
 23  V23     5973 non-null   float64
 24  V24     5973 non-null   float64
 25  V25     5973 non-null   float64
 26  V26     5973 non-null   float64
 27  V27     5973 non-null   float64
 28  V28     5973 non-null   float64
 29  Amount  5973 non-null   float64
 30  Class   5973 non-null   float64
dtypes: float64(30), int64(1)
memory usage: 1.4 MB
```

```
class_counts = df['Class'].value_counts()
print(class_counts)
```

```
0.0    5970
1.0       3
Name: Class, dtype: int64
```

```
legit = df[df['Class'] == 0]
fraud = df[df['Class'] == 1]
```

```
print(legit.shape)
print(fraud.shape)
```

```
(5970, 31)
(3, 31)
```

```
legit.Amount.describe()
```

```
count    5970.000000
mean       64.965707
std       192.429839
min         0.000000
25%         4.450000
50%        15.620000
75%        56.485000
max      7712.430000
Name: Amount, dtype: float64
```

```
fraud.Amount.describe()
```

```
count      3.000000
mean     256.310000
std      264.880121
min        0.000000
25%      119.965000
50%      239.930000
75%      384.465000
max      529.000000
Name: Amount, dtype: float64
```

```
df.groupby('Class').mean()
```

Class	Time	V1	V2	V3	V4	V5	V6	V7
0.0	2677.40201	-0.264965	0.285625	0.844580	0.102656	0.000958	0.195420	0.018542
1.0	1780.00000	-2.553039	0.184644	-0.293711	2.872264	0.005330	-0.855718	-0.549831

2 rows × 30 columns

```
legit_sample = legit.sample(n=492)
```

```
new_dataset = pd.concat([legit_sample, fraud], axis=0)
```

```
new_dataset.describe()
```

	Time	V1	V2	V3	V4	V5	V6
count	495.000000	495.000000	495.000000	495.000000	495.000000	495.000000	495.000000
mean	2658.416162	-0.301563	0.268080	0.880487	0.160723	0.033159	0.154765
std	1850.578338	1.424552	1.314983	1.027103	1.478113	1.173366	1.273941
min	0.000000	-12.168192	-15.732974	-3.025031	-3.650993	-3.288328	-7.465603
25%	940.500000	-1.032926	-0.273558	0.296381	-0.722867	-0.616035	-0.646636
50%	2462.000000	-0.442848	0.336488	0.920624	0.202238	-0.119393	-0.142256
75%	3949.000000	1.081892	0.950100	1.600173	1.171983	0.480037	0.579275
max	6644.000000	1.573518	4.621149	3.220953	3.997906	10.658654	4.417143

8 rows × 31 columns

```
new_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 495 entries, 3363 to 4920
Data columns (total 31 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Time    495 non-null    int64  
 1   V1      495 non-null    float64
 2   V2      495 non-null    float64
 3   V3      495 non-null    float64
 4   V4      495 non-null    float64
 5   V5      495 non-null    float64
 6   V6      495 non-null    float64
 7   V7      495 non-null    float64
 8   V8      495 non-null    float64
 9   V9      495 non-null    float64
 10  V10     495 non-null    float64
 11  V11     495 non-null    float64
 12  V12     495 non-null    float64
 13  V13     495 non-null    float64
 14  V14     495 non-null    float64
 15  V15     495 non-null    float64
 16  V16     495 non-null    float64
 17  V17     495 non-null    float64
 18  V18     495 non-null    float64
 19  V19     495 non-null    float64
 20  V20     495 non-null    float64
 21  V21     495 non-null    float64
 22  V22     495 non-null    float64
 23  V23     495 non-null    float64
 24  V24     495 non-null    float64
 25  V25     495 non-null    float64
 26  V26     495 non-null    float64
 27  V27     495 non-null    float64
 28  V28     495 non-null    float64
 29  Amount  495 non-null    float64
 30  Class   495 non-null    float64
dtypes: float64(30), int64(1)
memory usage: 123.8 KB
```

```
from sklearn.model_selection import train_test_split
```

```
x = new_dataset.drop(columns='Class', axis=1) # variable dependant
y = new_dataset['Class'] # Variable independant


print(y)
```

```
3363    0.0
2577    0.0
4155    0.0
1892    0.0
268     0.0
       ...
2743    0.0
696     0.0
541     1.0
623     1.0
4920    1.0
Name: Class, Length: 495, dtype: float64
```

```
print(x)
```

```
      Time        V1        V2        V3        V4        V5        V6  \
3363  2910 -0.300381  0.570978  0.716708 -2.214918  0.433500 -1.064170
2577  2111 -0.424596  0.647863  1.203465 -2.003843  0.168672 -1.229614
4155  3748  1.310845 -0.833391 -0.967631 -1.578280  1.412067  3.291606
1892  1462  1.294446  0.499666 -0.792481  0.527639  0.481296 -0.622905
268    190  0.075750  1.026986  0.834891  1.208913  0.107554  0.465977
...    ...       ...       ...       ...       ...       ...       ...
2743  2280  1.065920  0.014067  1.504940  1.410945 -0.951239 -0.011478
696    525 -0.755011 -0.517761  1.760091 -0.654206 -0.039143 -0.492847
541    406 -2.312227  1.951992 -1.609851  3.997906 -0.522188 -1.426545
623    472 -3.043541 -3.157307  1.088463  2.288644  1.359805 -1.064823
4920  4462 -2.303350  1.759247 -0.359745  2.330243 -0.821628 -0.075788

            V7        V8        V9  ...       V20       V21       V22  \
3363  1.044465 -0.296050  1.140068  ... -0.048111 -0.013581  0.265651
2577  1.079705 -0.305870  0.859738  ... -0.034335  0.105561  0.700966
4155 -1.108213  0.698196  0.354590  ...  0.317460 -0.249724 -0.922984
1892  0.249390 -0.095526 -0.303471  ... -0.064173 -0.145205 -0.413296
268  -0.330959 -1.619873 -0.436604  ... -0.258697  1.490696 -0.532414
...        ...       ...       ...  ...       ...       ...       ...
2743 -0.536191  0.122136  0.662543  ... -0.096398  0.016858  0.340324
696  -0.047345  0.118936  0.734444  ...  0.239908 -0.049608 -0.200904
541  -2.537387  1.391657 -2.770089  ...  0.126911  0.517232 -0.035049
623   0.325574 -0.067794 -0.270953  ...  2.102339  0.661696  0.435477
4920  0.562320 -0.399147 -0.238253  ... -0.430022 -0.294166 -0.932391

           V23       V24       V25       V26       V27       V28  Amount
```

```
3363 -0.379911 -0.494418  0.129384 -0.781942  0.227018 -0.025715    1.00
2577 -0.222024  0.396460 -0.054898 -0.828822  0.253524 -0.014524    1.00
4155  0.039953  0.922425  0.339828 -0.513378 -0.028782  0.023597   88.81
1892 -0.189173 -0.594400  0.638510  0.403843 -0.041412  0.017249    0.76
268  -0.179765 -0.422717  0.774770 -0.172429  0.265078  0.230016   20.00
...        ...       ...       ...       ...       ...       ...     ...
2743  0.072357  0.638504  0.311529 -0.416504  0.097206  0.043506   12.99
696   0.268931  0.108087 -0.468660  0.729549 -0.017462  0.077163   79.54
541  -0.465211  0.320198  0.044519  0.177840  0.261145 -0.143276    0.00
623   1.375966 -0.293803  0.279798 -0.145362 -0.252773  0.035764  529.00
4920  0.172726 -0.087330 -0.156114 -0.542628  0.039566 -0.153029  239.93

[495 rows x 30 columns]
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2, random_state=2)
```

```
print(x.shape, x_train.shape, x_test.shape)
```

```
(495, 30) (396, 30) (99, 30)
```

```
from sklearn.linear_model import LogisticRegression # import the necessary librarie
```

```
model = LogisticRegression()
model.fit(x_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Converge
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
x_train_prediction = model.predict(x_train)
```

```
x_test_prediction = model.predict(x_test)
```

```
from sklearn.svm import SVC
```

```
svm_model = SVC()
svm_model.fit(x_train, y_train)
```

▾ SVC

SVC()

```
svm_x_train_prediction = svm_model.predict(x_train)
```

```
svm_x_test_prediction = svm_model.predict(x_test)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_model = RandomForestClassifier()
rf_model.fit(x_train, y_train)
```

▾ RandomForestClassifier

RandomForestClassifier()

```
rf_x_train_prediction = rf_model.predict(x_train)
```

```
rf_x_test_prediction = rf_model.predict(x_test)


from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix


lr_accuracy = accuracy_score(y_test, x_test_prediction)
lr_precision = precision_score(y_test, x_test_prediction)
lr_recall = recall_score(y_test, x_test_prediction)
lr_f1 = f1_score(y_test, x_test_prediction)
lr_confusion = confusion_matrix(y_test, x_test_prediction)

# Evaluate SVM
svm_accuracy = accuracy_score(y_test, svm_x_test_prediction)
svm_precision = precision_score(y_test, svm_x_test_prediction)
svm_recall = recall_score(y_test, svm_x_test_prediction)
svm_f1 = f1_score(y_test, svm_x_test_prediction)
svm_confusion = confusion_matrix(y_test, svm_x_test_prediction)

# Evaluate Random Forest
rf_accuracy = accuracy_score(y_test, rf_x_test_prediction)
rf_precision = precision_score(y_test, rf_x_test_prediction)
rf_recall = recall_score(y_test, rf_x_test_prediction)
rf_f1 = f1_score(y_test, rf_x_test_prediction)
rf_confusion = confusion_matrix(y_test, rf_x_test_prediction)
```

```
print("Logistic Regression:")
print(f"Accuracy: {lr_accuracy}")
print(f"Precision: {lr_precision}")
print(f"Recall: {lr_recall}")
print(f"F1 Score: {lr_f1}")
print(f"Confusion Matrix:\n{lr_confusion}")

print("\nSVM:")
print(f"Accuracy: {svm_accuracy}")
print(f"Precision: {svm_precision}")
print(f"Recall: {svm_recall}")
print(f"F1 Score: {svm_f1}")
print(f"Confusion Matrix:\n{svm_confusion}")

print("\nRandom Forest:")
print(f"Accuracy: {rf_accuracy}")
print(f"Precision: {rf_precision}")
print(f"Recall: {rf_recall}")
print(f"F1 Score: {rf_f1}")
print(f"Confusion Matrix:\n{rf_confusion}")
```

```
Logistic Regression:
Accuracy: 0.98989898989899
Precision: 1.0
Recall: 0.5
F1 Score: 0.6666666666666666
Confusion Matrix:
[[97  0]
 [ 1  1]]

SVM:
Accuracy: 0.9797979797979798
Precision: 0.0
Recall: 0.0
F1 Score: 0.0
Confusion Matrix:
[[97  0]
 [ 2  0]]

Random Forest:
Accuracy: 0.9797979797979798
Precision: 0.0
Recall: 0.0
F1 Score: 0.0
Confusion Matrix:
[[97  0]
 [ 2  0]]
```

What is Logistic Regression technique?

Logistic Regression serves as a widely utilized binary classification technique in the field of credit card fraud detection. Specifically, it determines the likelihood of a credit card transaction being fraudulent within this particular context. By utilizing a logistic function, the algorithm transforms a linear combination of various input features (such as transaction amount, time, and location) into a probability range from 0 to 1. With the aid of a predetermined threshold, transactions can then be classified as fraudulent or non-fraudulent based on their predicted probabilities.

During the training phase, the model acquires knowledge about the most suitable weights for each feature by undergoing an optimization process. This process involves modifying the logistic function to accurately align with the labeled historical data.Logistic Regression is a transparent and intelligible statistical method that offers a comprehensive insight into the influence of specific characteristics on prediction outcomes. This method proves to be highly efficient, especially when there is a linear relationship between features and the log-odds of fraud.

Once the training process is complete, the model is utilized to assess newly acquired transactions, offering a likelihood score for each transaction. By establishing a predetermined threshold, for example, 0.5, transactions can be categorized as either fraudulent or non-fraudulent. Logistic Regression is particularly suitable for situations in which simplicity and comprehensibility are crucial, rendering it an invaluable asset in the realm of credit card fraud detection.

Data collection ,training and evaluation phase are same as above mention algorithm.

DATASET:

The dataset used for training and testing the model from Kaagle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud).

The dataset contains transactions made by credit cards in September 2013 by European cardholders.

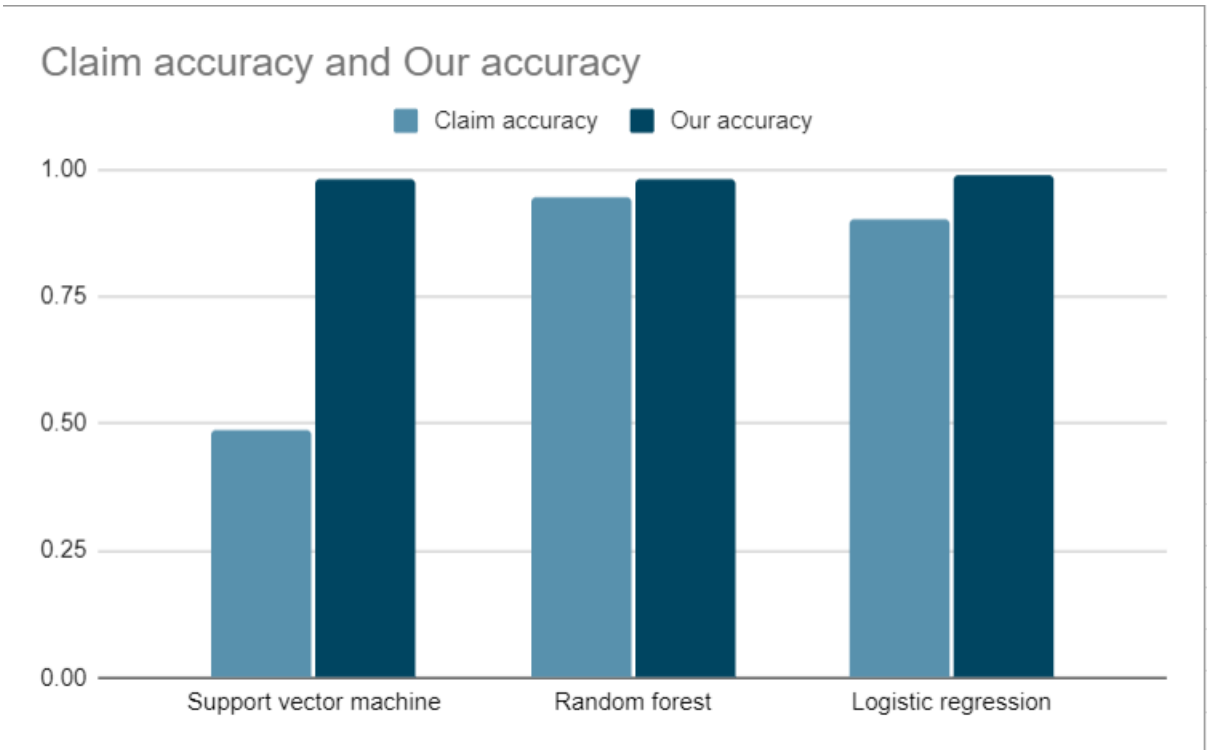
This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

Accuracy for our testing:

Algorithm	Claim accuracy	Our accuracy
Support vector machine	**0.4873096446700508**	**0.9797979797979798**
Random forest	**0.9441624365482234**	**0.9797979797979798**
Logistic regression	**0.9035532994923858**	**0.98989898989899**



OTHER METHODOLOGY:

The methodology section delineates the systematic approach undertaken to investigate fraud detection in online transactions using machine learning algorithms. This comprehensive framework encompasses data acquisition, preprocessing, model selection, training, and evaluation, ensuring methodological rigor and reproducibility in the study.

1. Data Acquisition:

The first pivotal step involves the acquisition of a representative dataset reflective of real-world online transactions. The dataset incorporates diverse transactional features, including temporal information, transaction amounts, user behaviors, and additional variables germane to fraud detection. Ethical considerations guide the data collection process, prioritizing user privacy and adherence to regulatory frameworks.

2. Data Preprocessing:

Data preprocessing is paramount in refining the acquired dataset for effective machine learning model training. This involves cleaning, transforming, and normalizing the data to address missing values, outliers, and inconsistencies. Feature engineering comes to the fore, as relevant variables are selected and crafted to encapsulate the intricacies of online transactional patterns.

3. Exploratory Data Analysis (EDA):

EDA delves into the dataset's statistical properties, uncovering patterns, trends, and potential correlations. Visualization techniques, such as histograms, scatter plots, and heatmaps, aid in elucidating the distribution of variables and identifying any discernible patterns indicative of fraudulent behavior.

4. Model Selection:

The choice of machine learning models is a critical decision, considering the intricacies of fraud detection. A comparative analysis of diverse algorithms is conducted, ranging from traditional classifiers like logistic regression to more sophisticated techniques such as random forests, support vector machines, and neural networks. The selection is driven by the need for a model capable of adapting to dynamic fraud patterns while maintaining interpretability and efficiency.

5. Feature Importance and Dimensionality Reduction:

Feature importance analysis is conducted to discern the variables contributing significantly to fraud detection. Dimensionality reduction techniques, including Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding (t-SNE), may be employed to distill the most salient features, enhancing model efficiency without compromising predictive power.

6. Model Training:

The selected machine learning models undergo rigorous training on the preprocessed dataset. Supervised learning models learn from labeled instances, distinguishing between legitimate and fraudulent transactions. Hyperparameter tuning is employed to optimize model performance, ensuring an adept adaptation to the intricacies of the data.

7. Cross-Validation and Model Evaluation:

Cross-validation techniques, such as k-fold cross-validation, validate the robustness of the trained models. The models are evaluated using appropriate metrics such as precision, recall, F1-score, and area under the Receiver Operating Characteristic (ROC) curve. Rigorous evaluation ensures the models effectively discriminate between genuine and fraudulent transactions.

8. Ensemble Methods and Hyperparameter Tuning:

Ensemble methods, combining multiple models to enhance predictive power, may be employed to fortify fraud detection capabilities. Hyperparameter tuning further refines the models, optimizing their configurations for heightened accuracy and adaptability.

9. Ethical Considerations:

Ethical considerations remain integral throughout the methodology. The responsible handling of sensitive transactional data, user privacy preservation, and adherence to legal frameworks underscore the ethical foundation of the study. Transparent communication and stringent data protection measures safeguard against any ethical dilemmas inherent in the research process.

10. Interpretability and Explainability:

Given the complexity of machine learning models, efforts are made to enhance model interpretability and explainability. Techniques such as LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations) are applied to elucidate how models reach decisions, fostering user trust and regulatory compliance.

CHAPTER - V : CONCLUSION

In conclusion, the implementation of machine learning algorithms for fraud detection in online transactions presents a significant advancement in enhancing the security and reliability of digital payment systems. Through this documentation, we have explored various machine learning techniques and methodologies employed in detecting fraudulent activities, including supervised learning, unsupervised learning, and hybrid approaches.

One of the primary advantages of utilizing machine learning for fraud detection is its ability to adapt and evolve with the changing nature of fraudulent behaviors. By analyzing vast amounts of transactional data in real-time, machine learning models can identify subtle patterns and anomalies indicative of fraudulent activities, thereby mitigating financial losses and protecting both consumers and businesses.

Throughout this documentation, we have highlighted the importance of feature engineering, model selection, and evaluation metrics in building robust fraud detection systems. Feature engineering plays a crucial role in extracting meaningful insights from raw transactional data, while careful consideration of model selection ensures the deployment of efficient and accurate fraud detection models.

Moreover, the use of appropriate evaluation metrics such as precision, recall, F1-score, and ROC-AUC allows for the quantitative assessment of model performance and the optimization of detection capabilities.

Furthermore, the integration of advanced techniques such as ensemble learning, anomaly detection, and deep learning holds promise for further improving the accuracy and efficacy of fraud detection systems. Ensemble methods, such as random forests and gradient boosting, leverage the collective wisdom of multiple models to achieve superior performance, while anomaly detection algorithms excel in identifying rare and previously unseen fraudulent patterns.

Additionally, deep learning architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), demonstrate remarkable potential in capturing complex hierarchical relationships within transactional data, leading to enhanced fraud detection capabilities.

However, despite the considerable progress made in the field of fraud detection using machine learning, several challenges and limitations persist. These include the imbalanced nature of transactional datasets, the presence of adversarial attacks, and the need for continuous model monitoring and adaptation to evolving fraud tactics. Addressing these challenges requires ongoing research and development efforts aimed at advancing algorithmic techniques, improving data quality, and implementing robust model deployment and maintenance strategies.

In conclusion, the integration of machine learning algorithms for fraud detection in online transactions represents a critical step towards enhancing the security and trustworthiness of digital payment systems. By leveraging the power of data-driven insights and predictive analytics, businesses and financial institutions can effectively combat fraudulent activities, safeguarding the integrity of online transactions and fostering greater confidence among consumers and stakeholders alike.

This conclusion summarizes the key points discussed in the documentation while emphasizing the significance of utilizing machine learning algorithms for fraud detection in online transactions.

CHAPTER - VI : FUTURE RESERCH

While significant progress has been made in leveraging machine learning algorithms for fraud detection in online transactions, several avenues for future research and innovation remain to be explored. The following are potential directions for advancing the field:

1. Enhanced Model Interpretability:

Despite the high accuracy achieved by machine learning models, their inherent complexity often hinders interpretability, making it challenging to understand the underlying factors driving fraudulent predictions. Future research could focus on developing interpretable machine learning techniques, such as explainable AI (XAI) methods, to provide transparent insights into model decisions and facilitate trust among users and stakeholders.

2. Adversarial Robustness:

As fraudsters continually adapt their tactics to evade detection, there is a growing need to enhance the robustness of machine learning models against adversarial attacks. Future research could explore techniques for improving model resilience to adversarial manipulation, including adversarial training, robust optimization, and adversarial detection mechanisms, to ensure the reliability and effectiveness of fraud detection systems in real-world scenarios.

3. Dynamic Model Adaptation:

Traditional machine learning models often struggle to adapt to evolving fraud patterns and shifting transactional dynamics over time. Future research could investigate adaptive learning approaches, such as online learning and reinforcement learning, to enable fraud detection systems to continuously update and refine their models in response to changing environments and emerging threats, thereby maintaining high detection accuracy and efficacy.

4. Privacy-Preserving Techniques: The collection and analysis of sensitive transactional data raise privacy concerns regarding the confidentiality and integrity of user information. Future research could explore privacy-preserving machine learning techniques, such as federated learning, differential privacy, and homomorphic encryption, to enable collaborative model training and inference while preserving data privacy and confidentiality, thereby addressing regulatory compliance requirements and protecting user privacy.

5. Integration of Multimodal Data:

In addition to transactional data, fraud detection systems could benefit from incorporating additional sources of information, such as user behavior patterns, device attributes, and contextual data, to enhance detection accuracy and robustness. Future research could focus on integrating multimodal data sources and leveraging advanced fusion techniques, such as deep learning-based multimodal fusion, to capture comprehensive insights into fraudulent activities and improve detection performance across diverse scenarios and use cases.

6. Cross-Industry Collaboration:

Collaboration and knowledge-sharing among stakeholders from different industries, including finance, cybersecurity, e-commerce, and telecommunications, can foster interdisciplinary research and innovation in fraud detection. Future research could explore collaborative approaches for sharing anonymized data, benchmarking detection algorithms, and developing standardized evaluation metrics to accelerate progress and enable the adoption of best practices across industries.

Future research in fraud detection for online transactions using machine learning algorithms holds immense potential for addressing emerging challenges, advancing technological capabilities, and enhancing the security and reliability of digital payment systems. By pursuing innovative research directions and fostering collaborative efforts, researchers and practitioners can contribute to the development of robust, scalable, and trustworthy fraud detection solutions that effectively combat fraudulent activities and safeguard the integrity of online transactions in an increasingly interconnected and dynamic digital landscape.

CHAPTER - VII : REFERENCE

https://www.researchgate.net/publication/342624204_Credit_card_fraud_detection_using_machine_learning_with_integration_of_contextual_knowledge

https://repository.rit.edu/cgi/viewcontent.cgi?article=12455&context=theses

https://cornerstone.lib.mnsu.edu/cgi/viewcontent.cgi?article=2167&context=etds

https://digitalscholarship.unlv.edu/cgi/viewcontent.cgi?article=4457&context=thesesdissertations

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

https://fraud-detection-handbook.github.io/fraud-detection-handbook

https://www.semanticscholar.org/paper/Credit-Card-Fraud-Detection-using-Deep-Learning-on-Pumsirirat-Yan/01be7624aa0e0251182593350a984411c2e5128a?p2df

https://ieeexplore.ieee.org/abstract/document/8424696

https://www.inderscienceonline.com/doi/abs/10.1504/IJEBANK.2020.114762

https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=4eb0dada0eb06cd4f354d0b7ecbe7664adb3badb

https://ieeexplore.ieee.org/abstract/document/8321781/

https://link.springer.com/chapter/10.1007/978-3-540-28651-6_119

https://www.academia.edu/download/55435740/733-1946-2-PB.pdf

https://ieeexplore.ieee.org/abstract/document/8316850/

KEYWORDS :

Credit card , fraud detection , ROC curve , Anomaly detection , machine learning , Fraud prevention , feature engineering , online transaction , algorithm based , Real time detection , financial fraud ,