Credit Card Scam Detection using Random Forest Algorithm

ABSTRACT — The aim of this research is sort out credit card fraud. The cost of fraud has dramatically increased over time. Money is taken from people by criminals by multiple means or fictitious information. The main aim of the scheme is devise machine learning models which would detect fake transactions preventing losses. And so it is very important to find out what can be done about such kind of deception.In this system one can find most essential elements required to determine illegal and criminal activities. The advances in technology hinder monitoring of illegal dealings including movement patterns. Release solutions that support growth in machine learning, artificial intelligence, and other IT-related technologies, and save some of the hard work of credit card verification. We first get the user's credit card data and split it into training and testing data, using decision trees and the random forest method. With this technique, we can examine more data, including the data already being used by the users. Continuing the process, we use some features that can be used to find the impact of spoofing when looking at the image structure of visual data.

Keywords - Random Forest, Crime, Credit Card Fraud, Decision Tree

I. SECTION
INTRODUCTION

The largest issue facing the modern world is credit card fraud, which demands immediate attention. Credit card fraud is also a major factor in identity theft. Considering the large number of financial transactions taking place every day in the global economy, checking credit card numbers is a difficult task. Looking at how the user spends money is the best way to discover fraud. Every day, scientists in different fields conduct new research. Researchers suggest using machine learning to solve this problem. Fraud detection is the process of identifying any people who are behaving suspiciously; when someone deviates from the normal paths, it becomes suspicious. This paper aimed at providing an algorithmic approach reliant on supervised learning in order to fight against credit card fraud. Trace algorithms are evolutionary algorithms which aim at finding improved answers gradually.There is a machine learning technique presented here for dealing with the problem.

It recognizes the traits and actions of other bank accounts through the use of unspecified "similar patterns". In the immense data, it is difficult to detect credit card fraud transactions; therefore, we propose a reduction method which will narrow down login data, find pairs of transactions that relate to other bank accounts and act on them if possible.To make sure that there is correctness in work request so as to avoid fraud and complicated calculations. A support vector machine is a classifier and pattern recognition technique. Such categorization scheme is aimed at predicting or dividing samples into two groups that can potentially be wrong.

II. SECTION
LITERATURE SURVEY

A. OVERVIEW

Credit card authority figures should manage to distinguish among unauthorized transactions and authorized ones so that their customers' accounts are safeguarded against incurring charges not approved by them. In order to avoid billing clients for purchases they did not make, this is essential. Since fraud causes large financial losses for many financial organizations, con artists are constantly looking for ways to commit crimes and take advantage of weaknesses. For this reason, banks that give credit cards must establish strong systems that detect fraud in order to avoid suffering losses. Different strategies, such as Neural Networks, Decision Trees, K-Nearest Neighbor algorithms, and Support Vector Machines, help in the identification of frauds.

B. LITERATURE RIVIEW

Identifying credit card system fraud using decision trees & support vector machine has both benefits in terms of cost and fast ending throughout the world. It makes use of automated machine learning tools that generate models by examining examples deposited in an Italian repository. This study uses a credit card system study to tackle a specific problem of financial losses resulting from illegitimate bank transactions.Many various methods are introduced for identifying credit card fraud (using neural networks, meta learning strategies, evolutionary algorithms, and HMMs). The detection of fraudulent activities within this particular system is made possible through an amalgamation of intelligent-data mining techniques as shown in decision trees or support vector machines carved out from within artificial intelligence concept. Such a hybrid approach helps cut down on financial waste mechanisms even further.

Mobile payment fraud is an unauthorized use of mobile gadgets to get monetary perks via theft of identity or credit card - Machine Learning Method for Mobile Payment Systems Financial Fraud Detection. The fraud of mobile payments is a quick-expanding problem as a result of the introduction of smartphones and online transition services. In the real world, mobile payment fraud identification is necessary to be very accurate because it results in financial loss. To identify fraud and handle huge financial data, our approach suggested a comprehensive process that would apply supervised and unsupervised machine learning methods to detecting mobile payment fraud.

III. SECTION

METHODOLOGY

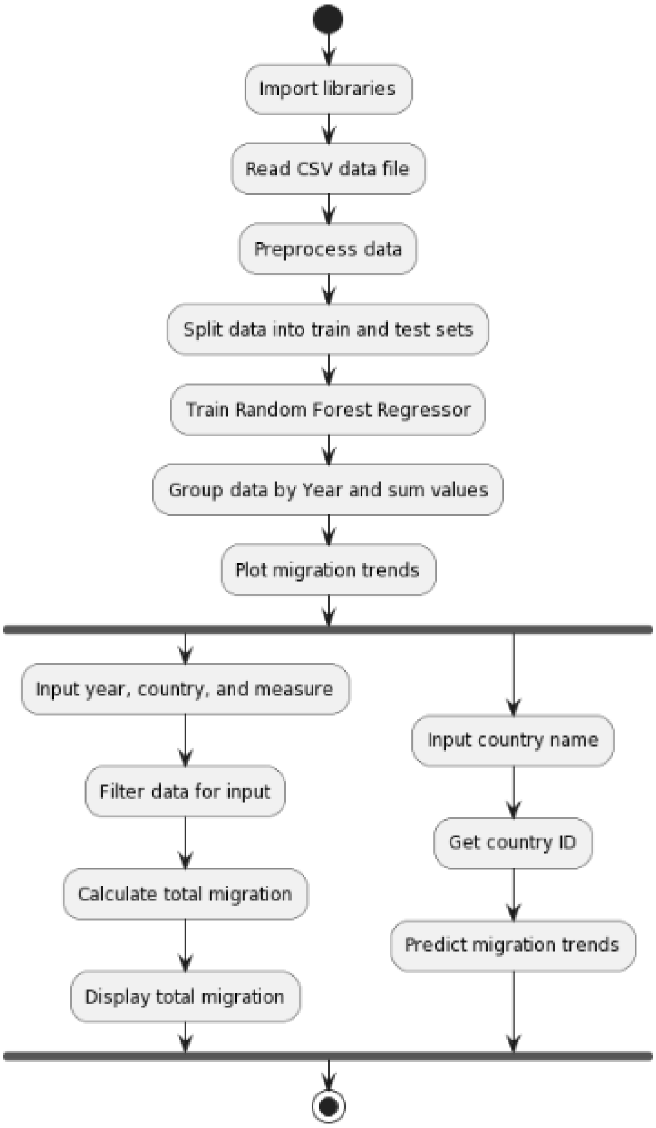
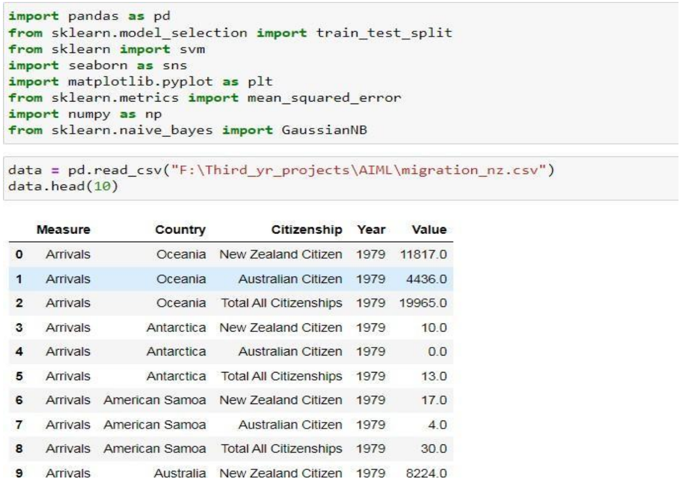


Fig.1 - Flowchart

A. DATA COLLECTION

Product reviews from credit card transaction records provided the analysis data. In this step, a particular subset of the total data set that is available for analysis is chosen. Machine learning projects usually start with a large amount of data, called labeled data, ideally with known target responses.



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn import svm
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
import numpy as np
from sklearn.naive_bayes import GaussianNB
```

```
data = pd.read_csv("F:\Third_yr_projects\AIML\migration_nz.csv")
data.head(10)
```

	Measure	Country	Citizenship	Year	Value
0	Arrivals	Oceania	New Zealand Citizen	1979	11817.0
1	Arrivals	Oceania	Australian Citizen	1979	4436.0
2	Arrivals	Oceania	Total All Citizenships	1979	19965.0
3	Arrivals	Antarctica	New Zealand Citizen	1979	10.0
4	Arrivals	Antarctica	Australian Citizen	1979	0.0
5	Arrivals	Antarctica	Total All Citizenships	1979	13.0
6	Arrivals	American Samoa	New Zealand Citizen	1979	17.0
7	Arrivals	American Samoa	Australian Citizen	1979	4.0
8	Arrivals	American Samoa	Total All Citizenships	1979	30.0
9	Arrivals	Australia	New Zealand Citizen	1979	8224.0

Fig. 2 - Importing libraries and Reading Dataset

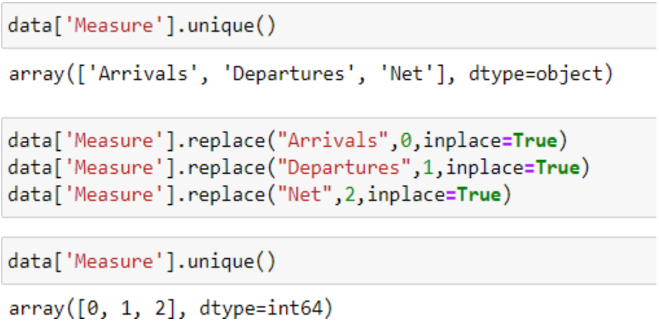
B. DATA PREPROCESSING

Pre-processing consists of three essential and frequently used steps:

- Formatting: This entails putting the information in an orderly fashion that makes it analytically possible. Usually, data files are formatted in accordance with specification of .csv format is strongly advised.

- cleansing: An important part of the workflow, data cleansing is a crucial component of data science. It includes things like fixing naming conventions, eliminating missing data, and streamlining complexity. For a lot of data scientists, cleansing data makes up about 80% of their effort.

- Sampling: This method entails taking and analyzing portions of huge datasets. A better knowledge of data behavior and patterns can be obtained by studying these subsets in a more manageable and integrated way.



```
data['Measure'].unique()
```

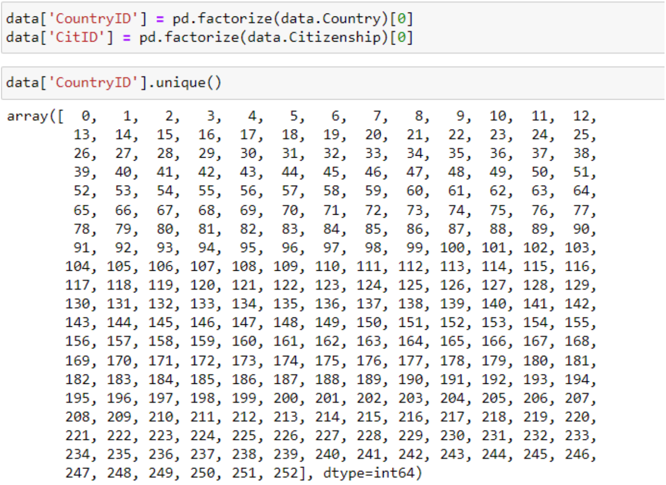
```
array(['Arrivals', 'Departures', 'Net'], dtype=object)
```

```
data['Measure'].replace("Arrivals",0,inplace=True)
data['Measure'].replace("Departures",1,inplace=True)
data['Measure'].replace("Net",2,inplace=True)
```

```
data['Measure'].unique()
```

```
array([0, 1, 2], dtype=int64)
```

Fig. 3 - Data preprocessing (replacing data values to integers)

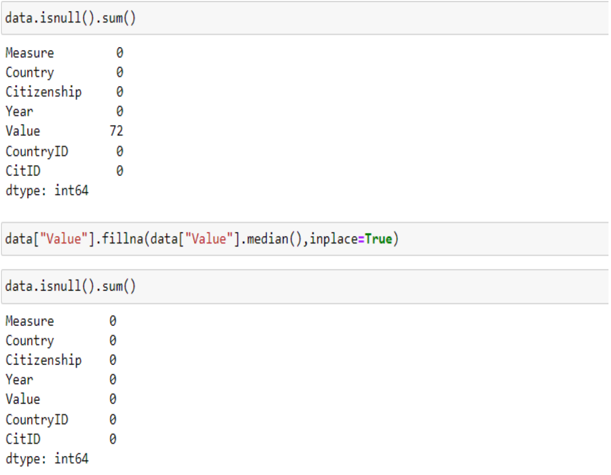


```
data['CountryID'] = pd.factorize(data.Country)[0]
data['CitID'] = pd.factorize(data.Citizenship)[0]
```

```
data['CountryID'].unique()
```

```
array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
        13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
        26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
        39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
        52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
        65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
        78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
        91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
       104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
       117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
       130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
       143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
       156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
       169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
       182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,
       195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
       208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
       221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233,
       234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
       247, 248, 249, 250, 251, 252], dtype=int64)
```

Fig. 4 - Data preprocessing (Adding column for evaluation)



```
data.isnull().sum()
```

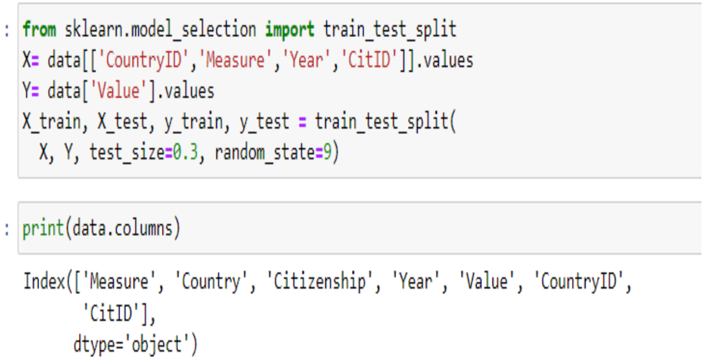
```
Measure         0
Country         0
Citizenship     0
Year            0
Value          72
CountryID       0
CitID           0
dtype: int64
```

```
data["Value"].fillna(data["Value"].median(),inplace=True)
```

```
data.isnull().sum()
```

```
Measure        0
Country        0
Citizenship    0
Year           0
Value          0
CountryID      0
CitID          0
dtype: int64
```

Fig. 5 - Data cleaning (Filling null values with median)



```
: from sklearn.model_selection import train_test_split
X= data[['CountryID','Measure','Year','CitID']].values
Y= data['Value'].values
X_train, X_test, y_train, y_test = train_test_split(
  X, Y, test_size=0.3, random_state=9)
```

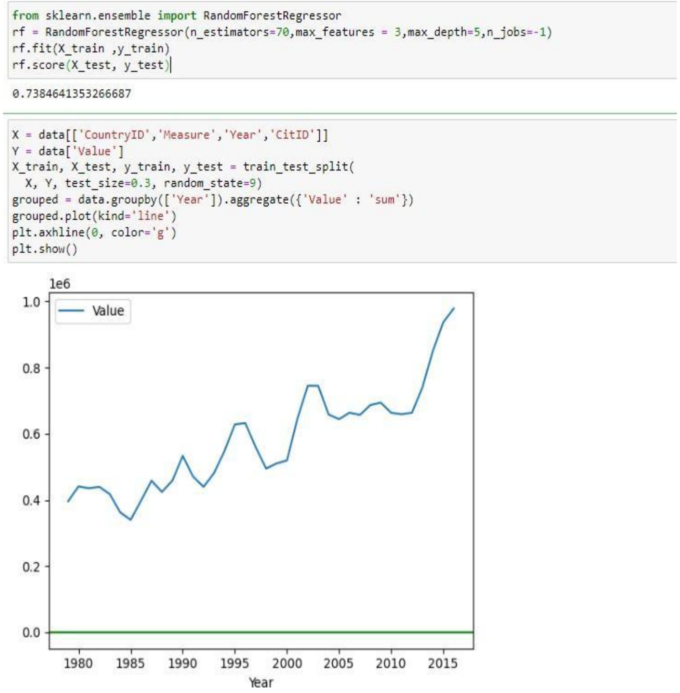
```
: print(data.columns)
```

```
Index(['Measure', 'Country', 'Citizenship', 'Year', 'Value', 'CountryID',
       'CitID'],
      dtype='object')
```

Fig. 6 - Splitting data into Train and test set

C. DATA VISUALIZATION

Data scientists can tell stories using information gained from visual and visual data analysis, which requires presenting data with pictures and visuals. Tableau is considered the best solution for this type of work because it has many features that make it easy to manage data and create beautiful results.



```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators=70,max_features = 3,max_depth=5,n_jobs=-1)
rf.fit(X_train ,y_train)
rf.score(X_test, y_test)
```

```
0.7384641353266687
```

```
X = data[['CountryID','Measure','Year','CitID']]
Y = data['Value']
X_train, X_test, y_train, y_test = train_test_split(
  X, Y, test_size=0.3, random_state=9)
grouped = data.groupby(['Year']).aggregate({'Value' : 'sum'})
grouped.plot(kind='line')
plt.axhline(0, color='g')
plt.show()
```

Fig. 7 - Test the model and predict the trend of migration overs years using line plot

D. FEATURE EXTRACTION

Removing features requires looking at behavior and patterns in the product being analyzed, in order to find relevant features for further testing and training. Afterward, these patterns are detected through classifier. The natural language toolkit is what we mostly use in Python deployment module. We employ recorded data in training.There are still some more labeled data left so that we can check on both of our models and check their performance. Some machine learning methods are applied for classifying already processed data among which random forest has been selected. The efficacy of these algorithms in text categorization tasks is well known.

E. MODEL EVALUATION

It is important for us to evaluate our models and see how they fit information or help us foresee what awaits us next. Evaluating models with same training data cannot be helpful as models may be too optimistic or even very complex but it’s wrong in doing so if one feels that way. Evaluate the model's performance using measurement techniques such as retention and competition to avoid overexposure. Often the results are presented graphically using categorized data.It is useful to consider the accuracy, which is the proportion of test forecasts that were correct. The forecast estimate can be calculated by dividing the total number of forecasts by the total number of observations.

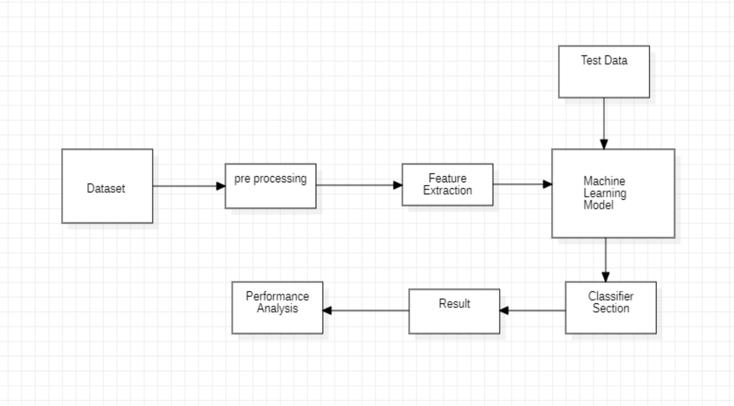


Fig. 8 - System Architecture

F. ALGORITHM

1) RANDOM FOREST

Random Forest is a supervised machine learning system with the concept of the combination of things. Again, numerous examples or variants of the same example aid the process of learning for making forecasts. This forms the basis of the random forest algorithm that is named so because it creates a forest from many decision trees. This approach is particularly versatile because it can be used in predicting so well as classifying.

Random Forest Algorithm Testing is done by follows:

Step 1 : Importing RandomForestRegressor.
Step 2 : Instantiate RandomForestRegressor.
Step 3 : Train the model.
Step 4 : Evaluate Model Performance by using R2 score. The higher the R2 score, the better the model fits the data.

2) ADVANTAGES OF RANDOM FOREST

- Total bias is decreased since each tree is trained independently using the data, and also boasts several trees, making the random forest approach non-biased.

- The algorithm is very robust. The entire algorithm is not affected if you add another data point.

- When one has both numerical and categorical attributes, it works well. Moreover, Random forest technique does well on instances where dataset has been improperly scaled or contains missing values. As a result, analyzing the data attributes facilitated our ability to come up with the exact percentage concerning fraud detection using Random Forest and Decision Tree algorithms.A confusion matrix is a table or summary of prediction outcomes that describes how a classifier performs on a set of test data where the true values are known. It simplifies classifying and illustrates the use of algorithms. Hence, much information regarding the nature of errors made by the classification model besides their errors can be used in computing most metrics measures of performance.The confusion matrix that is depicts the trained and testing data reveals:

True Positive(TP) : True Positive which implies genuine data on clients who have been fraudulently targeted used for training and forecasted accurately.

True Negative(TN) : The information which nobody expected and does not belong to the data with fraudulent origins is referred to as TN(True Negative).

False Positive(FP) : FP is often misinterpreted as False Positive, though it has no chances of fraudulent usage.

False Negative(FN) : In the case of FN, false negatives are not anticipated; however, there is an actual possibility that the data has been manipulated by fraudsters.

IV. CONCLUSION

In summary, the project provides a comprehensive assessment of New Zealand's immigration system, using machine learning and data visualization to provide a better understanding. We ensure that the data is suitable for modeling by pre-processing the dataset and coding categorical variables. Using a random forest regressor, we developed a prediction model that can predict the transition pattern based on historical data. Visualizations including line and bar charts provide a clear understanding of migration over the years and facilitate analysis. Additionally, user interaction features improve the usability of the project, allowing users to search for specific transit events and retrieve the entire number of transit locations by year and country. Overall, the project not only introduces the use of machine learning in migration analysis but also provides practical tools for policymakers, researchers, and participants seeking to understand and predict intercountry migration. By using a large number of training data sets, one finds that theRandom Forest algorithm perhaps does better. However, extensive testing and implementation time is affected. But, some additional preprocessing stages might be necessary. Our next study intends to use tools that are at the leading edge like deep learning, artificial intelligence, machine learning, to attempt solving credit card fraud through using a software application.

V. REFERENCES

1. With a logical order, this research has the objective of evaluating the effectiveness of different machine learning techniques in detecting credit card fraud, under the title “Systematic Review and Meta-analysis of Credit Card Fraud Detection Using Machine Learning Techniques” authored by L.M. Shantinath and M. Priya.

2. “This paper systematically explores recent trends and challenges associated with detection of credit card frauds by means of machine learning” – N. M. Mane and M. D. Ingle.

3. This book chapter written by Chandrashekar K. and Padmanabhan V. deals with a variety of machine-learning methods (both supervised as well as unsupervised) which are used to uncover credit card frauds.

4. "Credit Card Fraud Detection: Modeling Realistic Scenarios and a Novel Learning Approach" by S. Bhattacharyya, R. Jaiswal, and Nasipuri: Here, we propose a new way of thinking about credit card fraud detection in which the learnability of computers is increased.

5. "The document on Credit Card Fraud Detection using Machine Learning Techniques," wrote A. Zareei, M. Salahi and H. Javadi, offers a comprehensive analysis of different methods used to detect fraud in credit card transactions including various algorithms and methods of data preparation.

6. "Unsupervised Learning Approaches for Anomaly Detection in Credit Card Transactions" by P. Gupta, R. Verma, and S. Khanna: This paper investigates unsupervised learning methods for anomaly detection in credit card transactions, focusing on their effectiveness in identifying fraudulent activities.

7. "Fraud Detection in Real-Time Credit Card Transactions Using Machine Learning" by A. Mittal, S. Kumar, and V. Gupta: This research explores real-time fraud detection techniques for credit card transactions, emphasizing the role of machine learning in preventing fraudulent activities.

8. "Deep Reinforcement Learning for Credit Card Fraud Detection in Online Transactions" by R. Gupta, S. Mishra, and A. Singh: This research investigates the application of deep reinforcement learning for credit card fraud detection in online transactions, aiming to improve detection efficiency and reduce false positives.

9. "Feature Engineering for Credit Card Fraud Detection: An Empirical Study" by S. Sharma, A. Chauhan, and R. Gupta: This empirical study examines various feature engineering techniques for credit card fraud detection, assessing their impact on model performance and interpretability.

10. "Adversarial Machine Learning for Robust Credit Card Fraud Detection" by N. Jain, R. Verma, and S. Singh: This paper explores the use of adversarial machine learning approaches to enhance the robustness of credit card fraud detection models against sophisticated adversarial attacks.