ONLINE PAYMENTS FRAUD DETECTION

1. INTRODUCTION

1.1 Overview

Online payment systems have become increasingly popular in recent years, as they provide a convenient and secure way for individuals and businesses to make transactions. However, with the rise in online transactions comes an increase in the number of fraud cases. Fraud detection is a crucial aspect of online payment systems, as it helps to protect both the consumers and the merchants from financial losses. In this project, we propose a fraud detection system for online payments that utilizes machine learning techniques to identify and prevent fraudulent transactions. Our system aims to increase the accuracy of fraud detection while minimizing the number of false positives, resulting in a more efficient and effective system for detecting and preventing fraud.

1.2 Purpose

The purpose of this project report is to present the design, implementation, and evaluation of an online payment fraud detection system using machine learning. The main objectives of this project are:

- To analyze and understand the various types of fraud that occur in online payment systems.

- To propose and design a machine learning-based fraud detection system that can accurately identify and prevent fraudulent transactions.

- To implement the proposed system and evaluate its performance using real-world data.

- To provide insights and recommendations for future work in online payment fraud detection using machine learning.

Overall, the goal of this project is to contribute to the field of online payment security by developing a robust and efficient fraud detection system that can help to protect both consumers and merchants from financial losses.

2. LITERATURE SURVEY

2.1 Existing Problem

One of the main problems with online payments is the increasing number of fraud cases. Fraudulent transactions can cause significant financial losses for both consumers and merchants. Some of the existing problems with online payment fraud detection include:

- Difficulty in identifying fraudulent transactions: With the large volume of online transactions, it can be challenging to manually identify and flag fraudulent transactions.

- High false positive rate: Traditional fraud detection methods often result in a high number of false positives, where legitimate transactions are incorrectly flagged as fraudulent.

- Lack of flexibility: Existing fraud detection systems are not always able to adapt to new types of fraud or changing patterns of fraud.

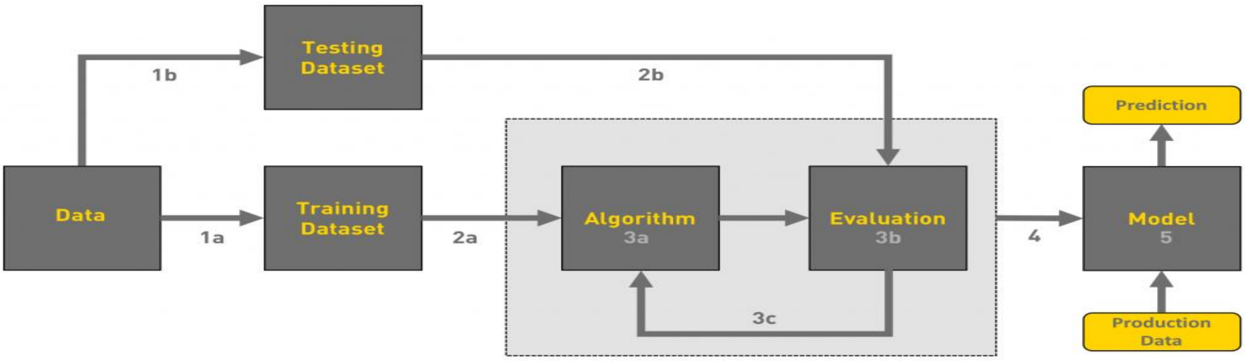
2.2 Proposed Solution

The proposed solution for this project is to design and implement a machine learning-based fraud detection system for online payments that addresses the existing problems and increase the accuracy of fraud detection while minimizing the number of false positives

The system will utilize state-of-the-art machine learning algorithms to improve the accuracy of fraud detection.The system will be designed to learn and adapt to new types of fraud and changing patterns of fraud. Overall, the proposed system aims to improve the effectiveness and efficiency of online payment fraud detection by utilizing advanced machine learning techniques and integrating multiple data sources to provide a more comprehensive view of each transaction.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware/Software Designing

Software Requirements:

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It was created by Guido van Rossum , and first released on February 20, 1991. Its high-level built in data structures, combined with dynamic typing and dynamic binding , make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution.Navigator allows you to launch common Python programs and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository.Conda is an open-source, crossplatform, package management system. For this project, we will be using Jupyter notebook and Spyder.

Jupyter Notebook

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access or can be installed on a remote server and accessed through the internet.

Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. It is released under the MIT license.

Flask

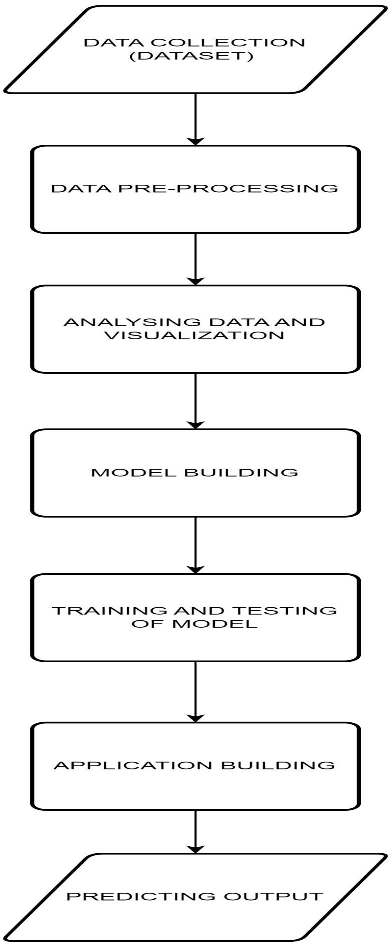
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

Hardware Requirements:
Operating System: Windows 7 or above
Processor: Intel Core i5 and above
RAM: 4Gb and above
Storage Space Required: 10gb and above

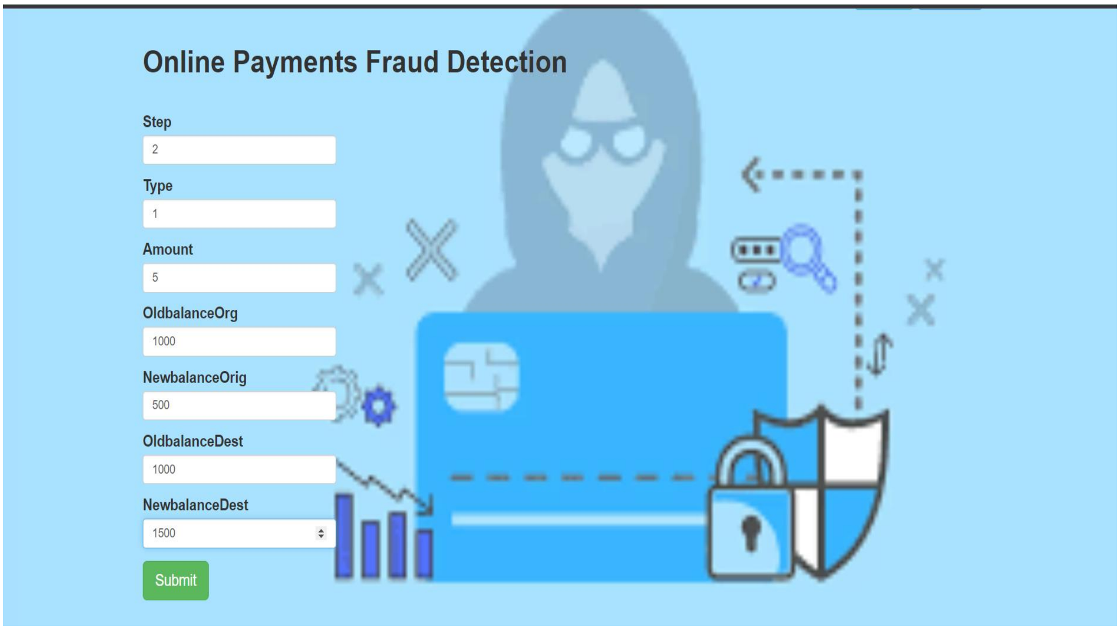
4. EXPERIMENTAL INVESTIGATIONS

The text data need to be organized before proceeding with the project. We will be using PS_20174392719_1491204439457_logs.csv dataset file to fetch the text data of training data. The datas are to be preprocessed in a way such that there is no empty field or outliers.We will create a function that uses the pre-trained model for predicting custom outputs. Then we have to test and train the model. After the model is build, we will be integrating it to a web application build in flask.

5. FLOWCHART


DATA COLLECTION (DATASET)
DATA PRE-PROCESSING
ANALYSING DATA AND VISUALIZATION
MODEL BUILDING
TRAINING AND TESTING OF MODEL
APPLICATION BUILDING
PREDICTING OUTPUT

6. RESULT



Online Payments Fraud Detection

Step: 2

Type: 1

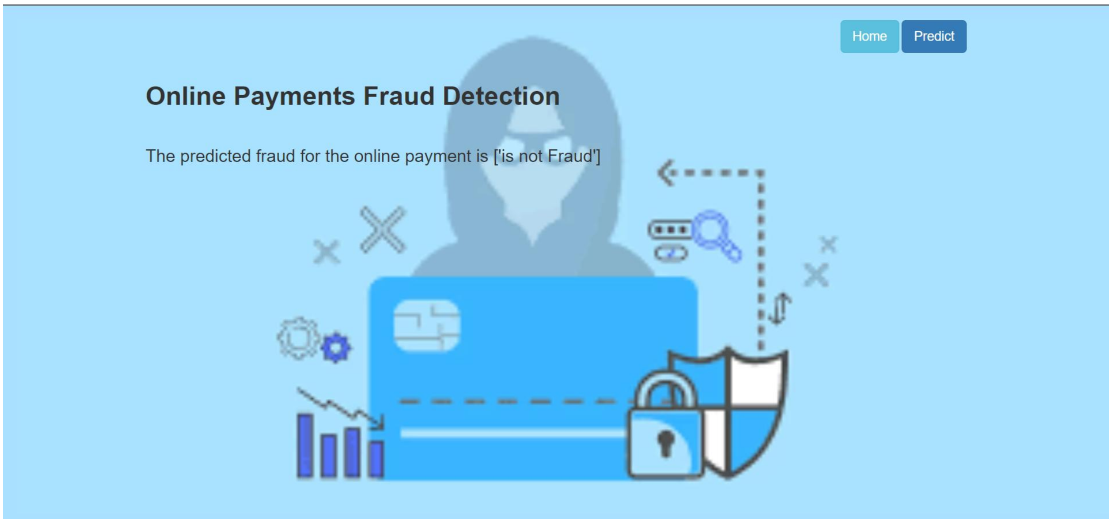
Amount: 5

OldbalanceOrg: 1000

NewbalanceOrig: 500

OldbalanceDest: 1000

NewbalanceDest: 1500



Online Payments Fraud Detection

The predicted fraud for the online payment is ['is not Fraud']

7. ADVANTAGES

- **Automated and efficient detection:** ML algorithms can analyze vast amounts of data and detect fraud patterns in real-time, reducing manual effort and errors.

- **Personalized risk assessment:** The system can take into account individual user behavior and purchase history to determine a personalized fraud risk score, enhancing fraud detection effectiveness.

- **Adaptability:** It can adapt to changing fraud patterns over time, continually improving the accuracy of fraud detection.

- **Scalability:** It can handle increasing amounts of data and transactions, making them suitable for organizations of all sizes.

- **Reduced costs:** Automated fraud detection reduces the need for manual investigations, thereby reducing operational costs.

DISADVANTAGES

- **Overreliance on technology:** Fraud Detection systems are not perfect, and relying solely on them may result in some fraudulent transactions going undetected.

- **False positives:** It may sometimes flag legitimate transactions as fraudulent, leading to inconvenience and loss of business for legitimate customers.

- **Data quality:** The quality of the data used to train the model affects their accuracy. Poor quality or biased data can result in incorrect fraud predictions.

- **Complexity:** ML systems can be complex and challenging to set up, requiring specialized skills and expertise**.**

8. APPLICATIONS

- **Healthcare fraud detection:** ML can be used to detect fraudulent billing practices in the healthcare industry, such as upcoding or phantom billing.

- **Telecommunication fraud detection:** ML algorithms can detect fraudulent activities in the telecommunication sector, such as unauthorized use of services or identity theft.

- **E-commerce fraud detection:** ML algorithms can detect suspicious behavior in e-commerce transactions, such as orders from high-risk countries or use of stolen credit card information.

9. CONCLUSION

This project was about predicting a online transcation as fraud or not.The proposed system had perfect accuracy while using different classification algorithms.classification algorithms used in this project are Decision tree, Random forest , xgboost Classifier.The project is then integrated with flask and also successfully deployed in IBM Cloud.

10. FUTURE SCOPE

Integration with biometrics: Integrating biometric authentication, such as facial recognition or fingerprint scanning, can improve the security of online transactions and reduce the risk of fraud.

Real-time monitoring: Improving real-time monitoring capabilities, for example by using edge computing, can enhance the effectiveness of fraud detection by analyzing data and making predictions in near real-time.

Big data analytics: Incorporating big data analytics can provide a more comprehensive view of transaction behavior and help identify emerging fraud trends.

Artificial Intelligence (AI) technologies: Integrating AI technologies such as deep learning or reinforcement learning can improve the accuracy and adaptability of fraud detection systems over time.

11. BIBLIOGRAPHY

- https://www.researchgate.net/publication/354937786_Online_Transaction_Fraud_Detection_System_Based_on_Machine_Learning

- https://thecleverprogrammer.com/2022/02/22/online-payments-fraud-detection-with-machine-learning/

12. APPENDIX

A. SOURCE CODE

APP.PY

```
from flask import Flask, render_template, request
import numpy as np
import pickle
import pandas as pd

model = pickle.load(open(r"D:/project/online payments fraud
detection/training/payments.pkl",'rb'))
app = Flask(__name__)
@app.route("/")
def about():
   return render_template('home.html')
@app.route("/home")
def about1():
   return render_template('home.html')
@app.route("/predict")
def home1():
   return render_template('predict.html')
@app.route("/pred", methods=['POST','GET'])
```

```
def predict():
   x = [[x for x in request.form.values()]]
   print(x)
   x = np.array(x)
   print(x.shape)
   print(x)
   pred = model.predict(x)
   print(pred[0])
   return render_template('submit.html', prediction_text=str(pred))
if __name__ == "__main__":
 app.run(debug=False)
```

HOME.HTML

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Home</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
    <style>
        body
        {
        background-image: url("data:image/png");
        background-size: cover;
        }
        h3.big
        {
        line-height: 1.8;
        }
    </style>
</head>
<body>
    <br>
    <div class="container">

        <div class="row">
            <div class="col-md-12 bg-light text-right">
                <a href="/home" class="btn btn-info btn-lg">Home</a>
                <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
```

```
         </div>
        </div>
        <center>
           <h1><strong>Online Payments Fraud Detection</strong></h1>
        </center>

        <h3 class="big"><em>The objective of this article is to predict
online payments fraud given the various parameters. This will be a
classification problem since the target or dependent variable is the
fraud(categorical values).The purpose of fraud of online paymetns are to
separate the available supply of potable online payments into classes
differing in superiority.
 We will be using classification algorithms such as Decision tree,
Random forest,  svm, and Extra tree classifier.  We will train and test the
data with these algorithms

        </em></h3><br>

     </div>

     <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></
script>
     <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js
"></script>
 </body>
 </html>
```

PREDICT.HTML

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Predict</title>
   <link                                                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
   <style>
      body
      {
```

```
    background-image: url("data:image/png");
    background-size: cover;
    }
    h3.big
    {
    line-height: 1.8;
    }
  </style>
</head>
<body>
  <br>
  <div class="container">

    <div class="row">
      <div class="col-md-12 bg-light text-right">
        <a href="/home" class="btn btn-info btn-lg">Home</a>
        <a href="/predict" class="btn btn-primary disabled btn-
lg">Predict</a>
      </div>
    </div> <br>
     <h1><strong>Online Payments Fraud
Detection</strong></h1><br>
     <h4>
     <form action="/pred", method="POST">
        <div class="form-group row">
          <div class="col-md-3">
            <label for="step">Step</label>
            <input type="number" class="form-control" name="step"
id="step"  placeholder="step: represents a unit of time where 1 step
equals 1 hour" required="required"/>
          </div>
        </div>
        <div class="form-group row">
          <div class="col-md-3">
            <label for="type">Type</label>
            <input type="number" class="form-control" name="type"
id="type" placeholder="type of online transaction"
required="required"/>
          </div>
        </div>
         <div class="form-group row">
           <div class="col-md-3">
             <label for="amount">Amount</label>
             <input type="number" class="form-control"
name="amount" min=5 max=15 step=0.000001 id="amount"
placeholder="the amount of the transaction" required="required"/>
           </div>
         </div>
         <div class="form-group row">
           <div class="col-md-3">
             <label for="oldbalanceOrg">OldbalanceOrg</label>
             <input type="number" class="form-control"
name="oldbalanceOrg" min=1000 max=7500000 step=0.01
id="oldbalanceOrg" placeholder="balance before the transaction"
required="required"/>
           </div>
         </div>
          <div class="form-group row">
            <div class="col-md-3">
              <label for="newbalanceOrig">NewbalanceOrig</label>
              <input type="number" class="form-control"
name="newbalanceOrig" min=0 max=500000 step=0.01
id="newbalanceOrig" placeholder="balance after the transaction"
required="required"/>
            </div>
          </div>
          <div class="form-group row">
            <div class="col-md-3">
              <label for="oldbalanceDest">OldbalanceDest</label>
              <input type="number" class="form-control"
name="oldbalanceDest" min=0 max=6500000 step=0.01
id="oldbalanceDest" placeholder="initial balance of recipient before the
transaction" required="required"/>
            </div>
          </div>
          <div class="form-group row">
            <div class="col-md-3">
              <label for="newbalanceDest">NewbalanceDest</label>
              <input type="number" class="form-control"
name="newbalanceDest" min=0 max=7500000 step=0.01
id="newbalanceDest" placeholder="the new balance of recipient after
the transaction" required="required"/>
            </div>
          </div>
          <button type="submit" class="btn btn-success btn-lg">Submit</button>
 </form>
        <br>
        </h4>
    </div>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</body>
</html>
```

SUBMIT.HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Output</title>
<linkrel="stylesheet"href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<style>
        body
        {
        background-image: url("data:image/png");
        background-size: cover;
        }
        h3.big
        {line-height: 1.8;}
</style>
</head>
<body>
    <br>
        <div class="container">
            <div class="row">
                <div class="col-md-12 bg-light text-right">
                    <a href="/home" class="btn btn-info btn-lg">Home</a>
                    <a href="/predict" class="btn btn-primary btn-lg">Predict</a>
</div>
```

```html
</div>
 <br>
<h1><strong>Online Payments Fraud Detection</strong></h1><br>
<h3>
The predicted fraud for the online payment is {{prediction_text}}
</h3>
</div>
</body>
</html>
```