

Project Title:

# ONLINE PAYMENTS FRAUD DETECTION

---

## 1 INTRODUCTION

### 1.1 Overview

The growth in internet and e-commerce appears to involve the use of online credit/debit card transactions. The increase in the use of credit / debit cards is causing an increase in fraud. The frauds can be detected through various approaches, yet they lag in their accuracy and its own specific drawbacks. If there are any changes in the conduct of the transaction, the frauds are predicted and taken for further process. Due to large amount of data credit / debit card fraud detection problem is rectified by the proposed method. 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.

We will be using classification algorithms such as Decision tree, Random forest, svm, and Extra tree classifier, xgboost Classifier. We will train and test the data with these algorithms. From this the best model is selected and saved in pkl format. We will be doing flask integration and IBM deployment.

### 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:

- Know fundamental concepts and techniques used for machine learning.
- Gain a broad understanding of data.
- Have knowledge of pre-processing the data/transformation techniques and some visualization concepts before building the model
- Learn how to build a machine learning model and tune it for better performance
- Know how to evaluate the model and deploy it using flask
- 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.

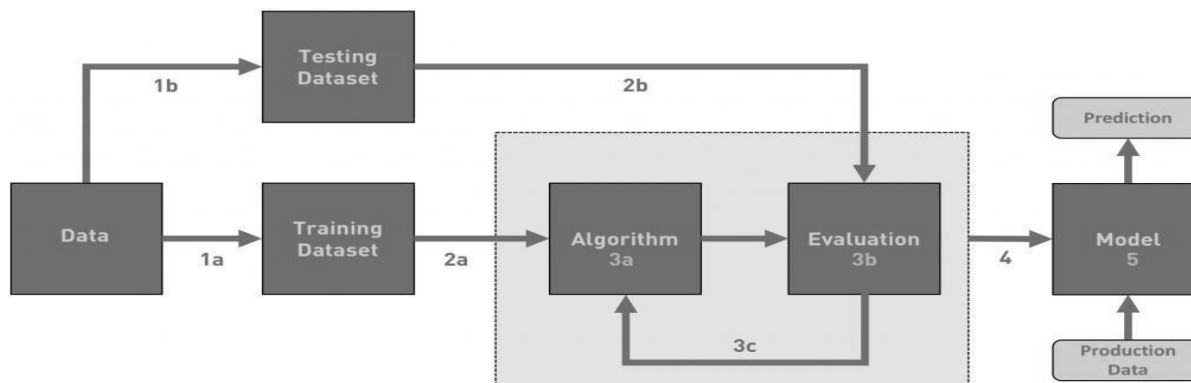
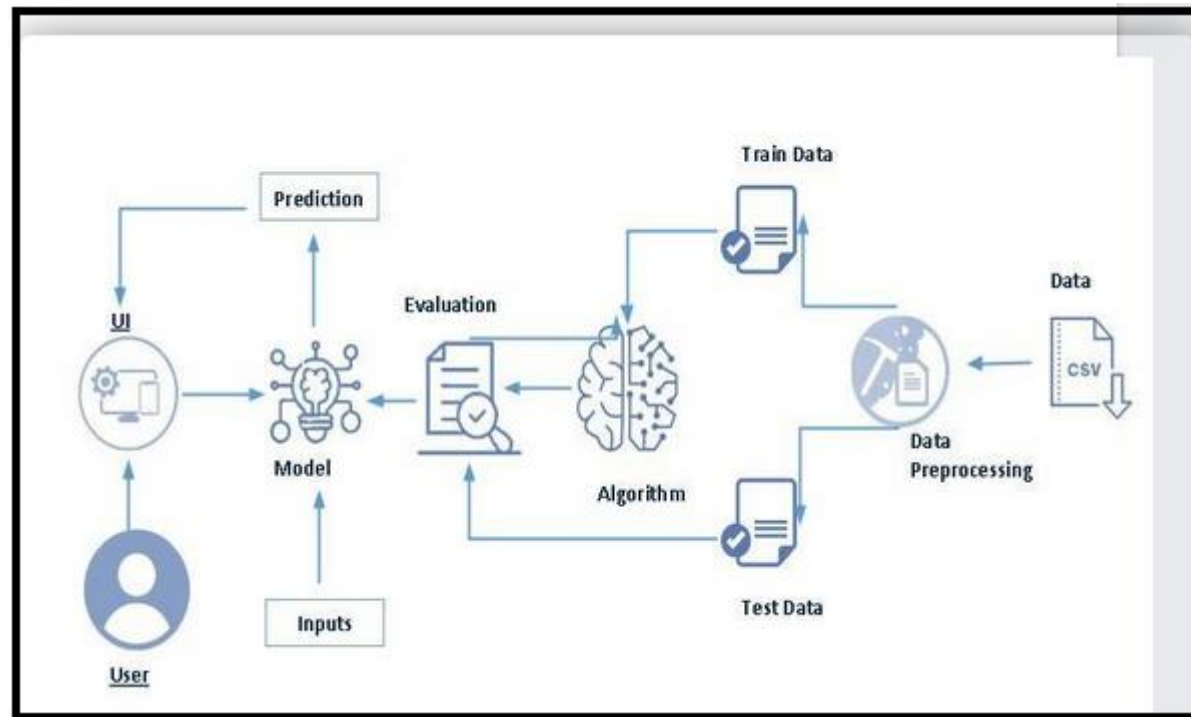
### 2.2 Proposed solution

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

##### 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.

## Python packages:

- Numpy
- Pandas
- scikit-learn
- matplotlib
- scipy
- pickle-mixin
- seaborn
- Flask

You must have prior knowledge of the following topics to complete this project.

- ML Concepts
- Supervised learning
- Unsupervised learning:

## 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

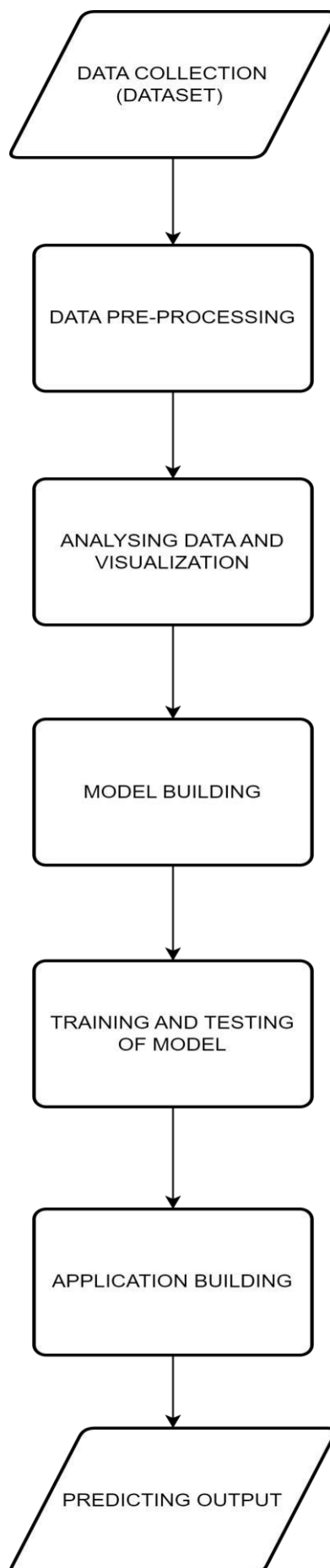
**RAM:** 4Gb and above

**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 kaggle dataset file to fetch the text data of training data. The data are to be preprocessed in a way such that there is no empty field or outliers and we will remove all the null values also. We will create a function that uses the pre-trained ml model for predicting custom outputs. Then we have to test and train the model. After the model is trained and then we will create an model using pickle, and then we will be integrating it to a web application build in flask.

## 5 FLOWCHART

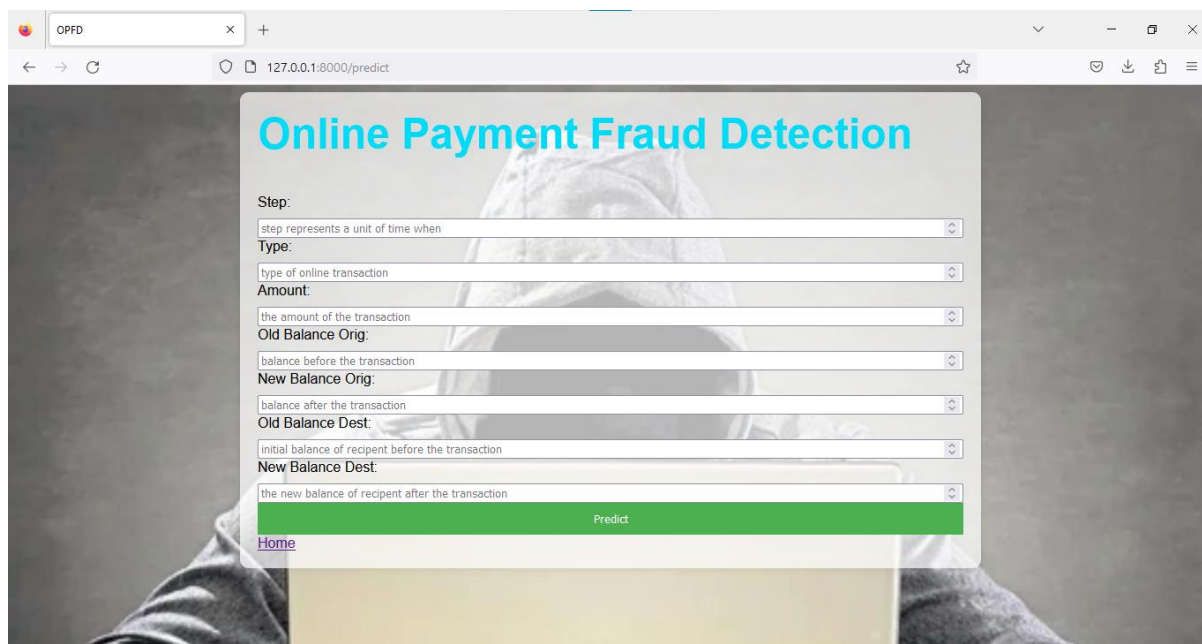


## 6 RESULT

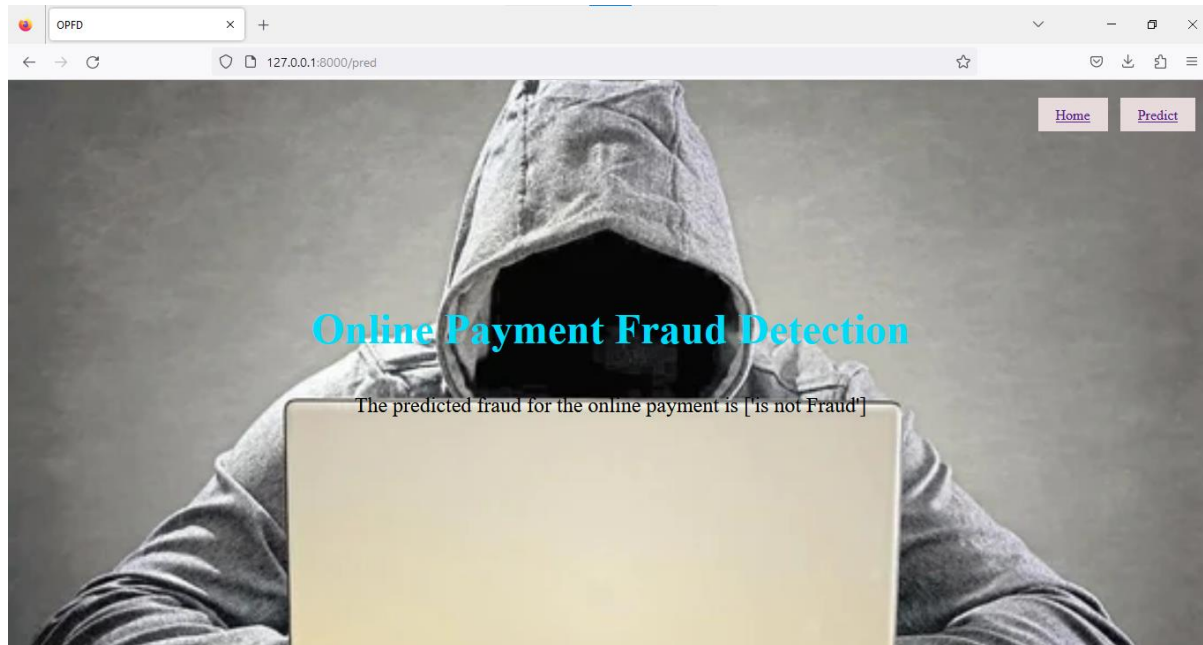
```
Anaconda Prompt - python app.py

(base) C:\Users\DELL>cd c:/Online_Payment_Fraud_Detection

(base) c:\Online_Payment_Fraud_detection>python app.py
C:\Users\DELL\anaconda3\lib\site-packages\sklearn\base.py:299: UserWarning: Trying to unpickle estimator SVC from version 1.0.2 when using version 1.2.1. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8000
Press CTRL+C to quit
* Restarting with watchdog (windowsapi)
```







## 7 ADVANTAGES & DISADVANTAGES

### 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 transaction 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.

## 9 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.

## 10 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/>

## 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"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=True, port=8000)

```

### home.html:

```

<!DOCTYPE html>
<html>
<head>
<title>OPFD</title>
<style>
    body {
        background-image: url({{ url_for('static',filename='online_img.jpeg')
}});
        background-size: cover;
        background-position: center;
        margin: 0;
        padding: 0;
    }

    .container {
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
        height: 100vh;
        text-align:center;
    }

    .heading {
        font-size: 48px;
        color: rgb(0, 219, 248);
        margin-bottom: 20px;
    }

    .content {
        font-size: 24px;
        color: rgb(255, 252, 252);

```

```

    margin-bottom: 40px;
}

.options {
    position: absolute;
    top: 20px;
    right: 20px;
}

.option-box {
    display: inline-block;
    background-color: rgb(231, 219, 219);
    padding: 10px 20px;
    margin-left: 10px;
}
</style>
</head>
<body>
<div class="container">
    <h1 class="heading">Online Payment Fraud Detection</h1>
    <p class="content">The objective of this article is to predict online
payment 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 paymnets 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.</p>
</div>

<div class="options">
    <div class="option-box" >Home</div>
    <div class="option-box"> <a href="/predict">Predict</a></div>
</div>
</body>
</html>

```

### Predict.html:

```

<!DOCTYPE html>
<html>
<head>
    <title>OPFD</title>
    <style>
        body {

```

```
background-image: url({{
url_for('static',filename='online_img.jpeg') }});
background-size: cover;
font-family: Arial, sans-serif;
}

.container {
max-width: 800px;
margin: 0 auto;
padding: 20px;
background-color: rgba(255, 255, 255, 0.7);
border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
font-size: 48px;
color: rgb(0, 219, 248);
margin-bottom: 40px;
margin-top: 0px;
}

form {
display: flex;
flex-direction: column;
}

label {
margin-bottom: 10px;
}

input[type="text"] {
width: 40%;
padding: 5px;
border-radius: 5px;
border: 1px solid #ccc;
margin-bottom: 10px;
}

button[type="submit"] {
padding: 10px;
background-color: #4CAF50;
color: white;
border: none;
cursor: pointer;
}
```

```
        button[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Online Payment Fraud Detection</h1>
        <form action="/pred" method="POST">
            <label for="step">Step:</label>
                <input type="number" name="step" min=1 max=100
placeholder="step represents a unit of time when" required>

            <label for="type">Type:</label>
                <input type="number" name="type" min=1 max=5
placeholder="type of online transaction" required>

            <label for="amount">Amount:</label>
                <input type="number" name="amount" placeholder="the amount
of the transaction" required>

            <label for="oldbalanceOrig">Old Balance Orig:</label>
                <input type="number" name="oldbalanceOrig"
placeholder="balance before the transaction" required>

            <label for="newbalanceOrig">New Balance Orig:</label>
                <input type="number" name="newbalanceOrig"
placeholder="balance after the transaction" required>

            <label for="oldbalanceDest">Old Balance Dest:</label>
                <input type="number" name="oldbalanceDest" placeholder="initial
balance of recipient before the transaction" required>

            <label for="newbalanceDest">New Balance Dest:</label>
                <input type="number" name="newbalanceDest" placeholder="the
new balance of recipient after the transaction" required>

            <button type="submit">Predict</button>
        </form>

        <div class="header">
            <a href="/">Home</a>
        </div>
    </div>
</body>
</html>
```

## Submit.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>OPFD</title>
  <style>
    body {
      background-image: url({{ url_for('static',filename='online_img.jpeg')
    }});
      background-size: cover;
      background-position: center;
      margin: 0;
      padding: 0;
    }

    .container {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      text-align:center;
    }

    .heading {
      font-size: 48px;
      color: rgb(0, 219, 248);
      margin-bottom: 20px;
    }

    .content {
      font-size: 24px;
      color: rgb(0, 0, 0);
      margin-bottom: 40px;
    }

    .options {
      position: absolute;
      top: 20px;
      right: 20px;
    }

    .option-box {
      display: inline-block;
      background-color: rgb(231, 219, 219);
      padding: 10px 20px;
```



```
        margin-left: 10px;
    }
</style>
</head>
<body>
    <div class="container">
        <h1 class="heading">Online Payment Fraud Detection</h1>
        <p class="content">The predicted fraud for the online payment is {{
prediction_text }}</p>
    </div>

    <div class="options">
        <div class="option-box" ><a href="/">Home</a></div>
        <div class="option-box"> <a href="/predict">Predict</a></div>
    </div>
</body>
</html>
```

---:THE END:---

---:THANK YOU:---