CHAPTER 1
ABSTRACT

ABSTRACT

An essential part of contemporary financial systems, the Online Payment Fraud Detection System was created to lessen the ever-growing threat of fraudulent activity in online transactions. By utilizing machine learning methods and algorithms, this system protects the integrity of online payment processes by identifying and preventing fraudulent transactions through the use of modern technologies and algorithms. An overview of the system, including its goals, workings, and vital role in guaranteeing safe online payments, is given in this description. The main goals of the system include detecting fraudulent transactions, but also minimizing interruptions for legal users, reducing false positives, and optimizing accuracy. Its approach entails ongoing learning from past data, allowing its models to be adjusted to account for changing fraud trends. Using a multifaceted methodology, the system examines user.

CHAPTER 2
INTRODUCTION

INRODUCTION

The way people make financial transactions has changed dramatically with the rise of e-commerce and online banking. Nevertheless, this ease of use has also drawn unscrupulous individuals who take advantage of holes in the digital payment system to perpetrate fraudulent activities. The Online Payment Fraud Detection System has become an indispensable instrument in the fight against this threat.

There is a bad side to the revolutionary surge of internet banking and e-commerce that has unquestionably made financial transactions easier for millions of people: a greater vulnerability to fraudulent activity. The ease of digital payment systems has been exploited by malicious actors who want to take advantage of system weaknesses. This poses a serious risk to the security and integrity of online financial transactions. As a result, the Online Payment Fraud Detection System was developed, which is a vital instrument in the continuous fight against this cyberthreat.

Operating on a continuous learning paradigm, the Online Payment Fraud Detection System modifies and enhances its algorithms in response to past data and new fraud tendencies. Because of its adaptive nature, the system is able to adapt to the ever-changing strategies used by malevolent entities, maintaining its effectiveness and agility. By using machine learning techniques, it becomes more adept at identifying novel patterns and small irregularities, strengthening its defences against sophisticated fraud efforts.

CHAPTER 3
LITERATURE SURVEY

LITERATURE SURVEY

Authors & Title of the paper	Type of Study	Tools used	Methodology used	Major Results
Fraud Detection for Online Transactions Using Random Forest by Jun Seok Kang, Hyunjung Shin, and Jongwon Kim. (2013)	The paper aims to achieve online payment fraud detection by leveraging text mining techniques and employing deep learning models constructed using LSTM neural networks. These models aim to accurately identify and classify instances of fraudulent activities, effectively filtering them out.	NTLK library, SpaCy, LSTM	An online payment fraud detection classifier is designed utilizing a dataset of 56,745 transactions. The dataset is preprocessed with NLTK, tokenized, and trained with LSTM on SpaCy word embeddings for 30,000 frequent words.	The developed model exhibits transactions as fraudulent or non-fraudulent with a precision of 94.49%, recall of 92.79%, and an overall accuracy score of 94.94%.
"Deep Learning Models for Online Fraud Detection" by M. H. Bhuyan, Dhruba Kumar Bhattacharyya, and J. K. Kalita. (2018)	The approach entails the utilization of ensemble models, specifically a convolutional neural network (CNN), bidirectional long short-term memory (LSTM), and bidirectional gated recurrent units (GRU), for the purpose of online payment fraud detection.	CNN, LSTM, GRU,	Two classifiers are implemented for online payment fraud detection: a binary classifier distinguishing between fraudulent and non-fraudulent transactions, and a multi-label classifier.	It achieves a F1-score of 0.828 for binary classification, distinguishing between fraudulent and non-fraudulent transaction
“A Study on Detection of Online Payment Frauds Based on Machine Learning Algorithms" by M. Nithya and R. Sumalatha. (2017)	Uses six machine learning algorithms and applies them to our data to address the challenge of online payment fraud detection. The goal is to identify the best machine learning algorithm based on evaluation metrics for classifying fraudulent and non-fraudulent transactions.	logistic regression, random forest, SVM classifier, naive Bayes, decision tree, and KNN classification	Logistic regression is selected for binary classification of online payment transactions as fraudulent or non-fraudulent, SVM is utilized for distinct fraud labels, integrating decision tree and random forest approaches. Additionally, Naïve Bayes and KNN classification are employed for independent labeled input data in identifying various types of online	The chosen final model for online payment fraud detection is the logistic regression model, demonstrating a maximum accuracy of 89.46% and the least possible hamming loss, which is 2.43%.

			payment fraud. The final classification is determined through decision tree voting.	

CHAPTER 4
METHODOLOGY

METHODOLOGY

The methodology entails compiling a diverse dataset of online payment transactions, preprocessing transaction data, extracting pertinent features, labeling transactions for fraud detection, selecting and training a deep neural network model, assessing its performance, interpreting indicators of fraudulent activities, comparing with baseline models, addressing ethical considerations, and discussing practical implementation and potential avenues for future research in the field of online payment fraud detection.

1. **Data Preparation**:
   Curate a comprehensive dataset of diverse online payment transactions, ensuring representation from various sources and transaction types, with a specific emphasis on fraud instances.

2. **Model Development**:
   Conduct data preprocessing, transforming transaction features through techniques like normalization and scaling. Experiment with different models, including Gradient Descent and Decision Trees, to identify the most effective architecture for online payment fraud detection.

3. **Integration into System**:
   Embed the trained fraud detection model into existing payment platforms, seamlessly integrating it into the transaction processing system to actively identify and flag potential fraudulent activities in real-time.

4. **Real-Time Prediction and Monitoring**:
   Deploy the integrated model for real-time prediction, continuously monitoring and analyzing incoming transactions. Implement mechanisms to trigger alerts or interventions for suspicious activities based on model predictions.

5. **Continuous Development**:
   Iteratively refine the fraud detection model by continuously updating it with new data, adapting to evolving patterns of fraudulent behavior. Employ techniques like hyperparameter tuning to enhance model performance over time.

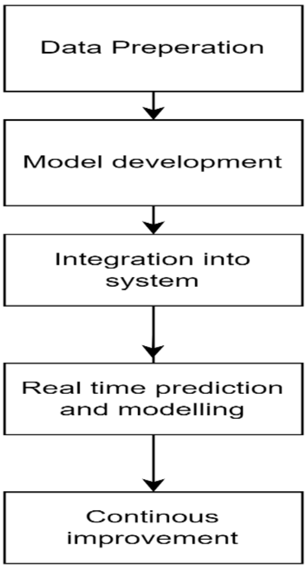


Figure 4.1: Methodology flow diagram

CHAPTER 5
IMPLEMENTATION

IMPLEMENTATION

In the realm of online payment fraud detection, a well-designed system employs Gradient Descent and Decision Tree models to discern anomalous patterns in transaction data. By preprocessing features like transaction amounts and timestamps, the models are trained to identify potentially fraudulent activities, showcasing their efficacy through accuracy, precision, and recall evaluations. The flexibility of Gradient Descent and the interpretability of Decision Trees synergize to fortify the system against fraudulent transactions.

1) Input Data (Features): The input data consists of information related to online payment transactions, features may include transaction amounts, timestamps, user information, etc.

2) Import Libraries: We import necessary libraries, such as NumPy, pandas, and scikit-learn, for data manipulation, analysis, and modeling, utilize scikit-learn for implementing the Gradient Descent model and Decision Tree.

3) Load and Explore Data: Load the payment transaction data into a Data Frame for convenient manipulation, conduct exploratory data analysis to understand the structure of the dataset and identify potential features for fraud detection.

4) Preprocess Data: Prepare the data for model training, split the dataset into features (X) and labels (y). If necessary, handle missing values, encode categorical variables, and scale numerical features.

5) Model architecture: Implement a Gradient Descent model using logistic regression for online payment fraud detection, utilize the logistic regression algorithm for binary classification, train the model using the gradient descent optimization algorithm, implement a Decision Tree model for comparison, train the Decision Tree model on the same dataset.

6) Training and analysis: For the Gradient Descent model, implement the gradient descent algorithm to iteratively update model parameters, analyze the training process using metrics such as training and validation loss.
   For the Decision Tree model, train the model using the fit method provided by scikit-learn, analyze the decision tree structure and feature importance.

7) Evaluation: Test the models on a random set of online payment transactions to assess classification accuracy, print and analyze key metrics such as accuracy, precision, recall, and loss, create a graph to visualize the variation of accuracy across different training epochs for the Gradient Descent model

CHAPTER 6
SYSTEM REQUIREMENTS AND SPECIFICATION

SYSTEM REQUIREMENTS AND SPECIFICATION

6.1 SYSTEM REQUIREMENT SPECIFICATION

System Requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's systemrequirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two- way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The testing and validation plans, and documentation plans, . It is important to note that an SRS contains functional and non-functional requirements only.

6.1.1 Hardware Specification

- Processor: intel core i3 or above
- Processor speed: 500Mhz or above
- RAM: 4GB or above

6.1.2 Software Requirement

- Python

6.1.3 Libraries and Tools

1. OpenCV
2. Tkinter
3. Numpy
4. Matplotlib
5. Jupyter Notebook

Software Specification:

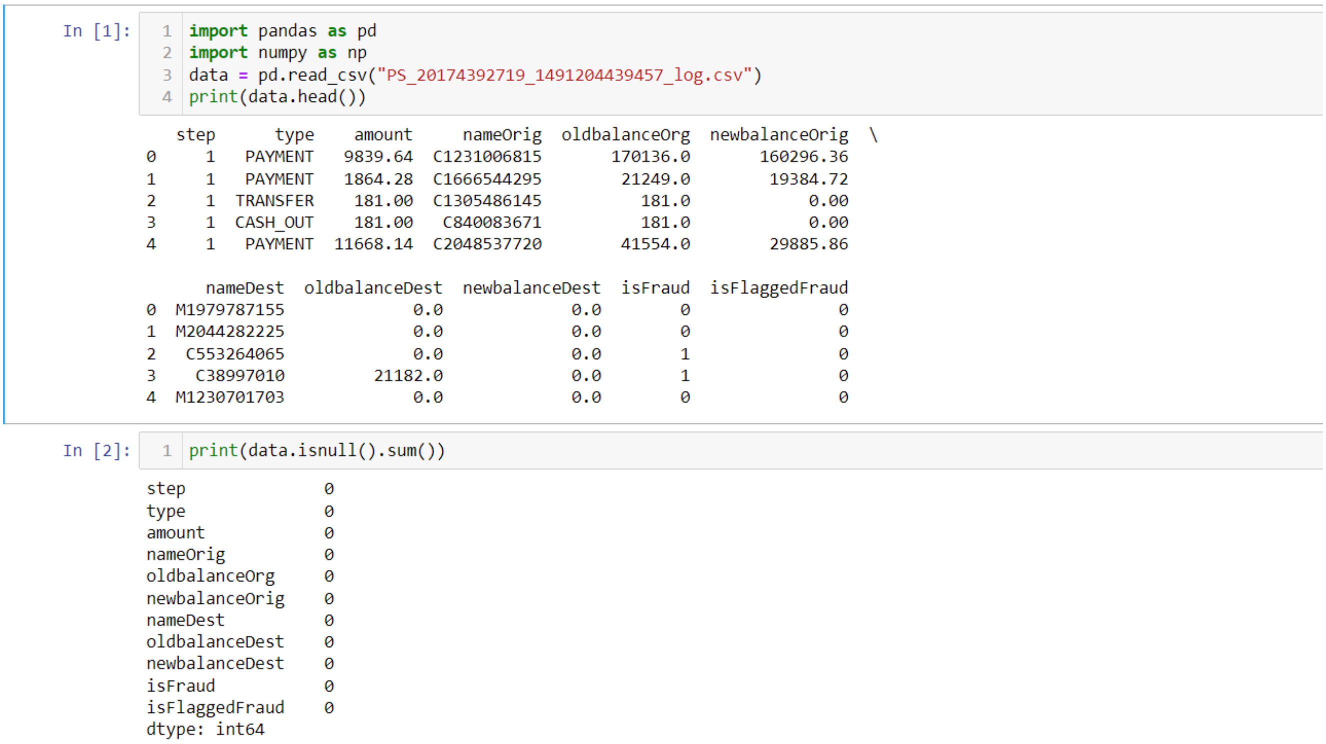
Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object- orientedapproach aim to help programmers write clear, logical code for small andlarge- scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a “batteries included” language due to its comparative standard library.

CHAPTER 7
RESULTS AND SNAPSHOTS

SNAPSHOTS



```
In [1]: import pandas as pd
        import numpy as np
        data = pd.read_csv("PS_20174392719_1491204439457_log.csv")
        print(data.head())
```

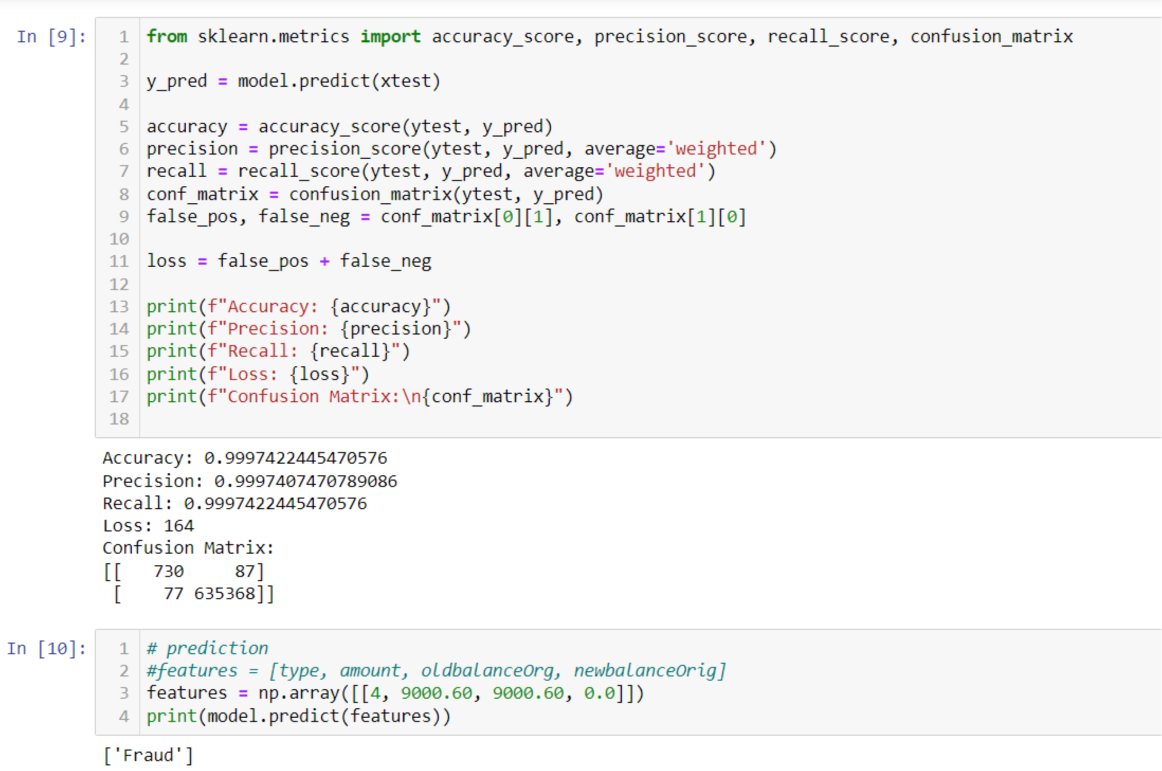
```
   step      type    amount     nameOrig  oldbalanceOrg  newbalanceOrig  \
0     1   PAYMENT   9839.64  C1231006815       170136.0       160296.36   
1     1   PAYMENT   1864.28  C1666544295        21249.0        19384.72   
2     1  TRANSFER    181.00  C1305486145          181.0            0.00   
3     1  CASH_OUT    181.00   C840083671          181.0            0.00   
4     1   PAYMENT  11668.14  C2048537720        41554.0        29885.86   

      nameDest  oldbalanceDest  newbalanceDest  isFraud  isFlaggedFraud  
0  M1979787155             0.0             0.0        0               0  
1  M2044282225             0.0             0.0        0               0  
2   C553264065             0.0             0.0        1               0  
3    C38997010         21182.0             0.0        1               0  
4  M1230701703             0.0             0.0        0               0  
```

```
In [2]: print(data.isnull().sum())
```

```
step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

Figure 7.1.1: Importing the necessary Python libraries



```
In [9]: from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix

        y_pred = model.predict(xtest)

        accuracy = accuracy_score(ytest, y_pred)
        precision = precision_score(ytest, y_pred, average='weighted')
        recall = recall_score(ytest, y_pred, average='weighted')
        conf_matrix = confusion_matrix(ytest, y_pred)
        false_pos, false_neg = conf_matrix[0][1], conf_matrix[1][0]

        loss = false_pos + false_neg

        print(f"Accuracy: {accuracy}")
        print(f"Precision: {precision}")
        print(f"Recall: {recall}")
        print(f"Loss: {loss}")
        print(f"Confusion Matrix:\n{conf_matrix}")
```

```
Accuracy: 0.9997422445470576
Precision: 0.9997407470789086
Recall: 0.9997422445470576
Loss: 164
Confusion Matrix:
[[   730     87]
 [    77 635368]]
```

```
In [10]: # prediction
         #features = [type, amount, oldbalanceOrg, newbalanceOrig]
         features = np.array([[4, 9000.60, 9000.60, 0.0]])
         print(model.predict(features))
```

```
['Fraud']
```

Figure 7.1.2: The correlation between the features of the data

RESULTS

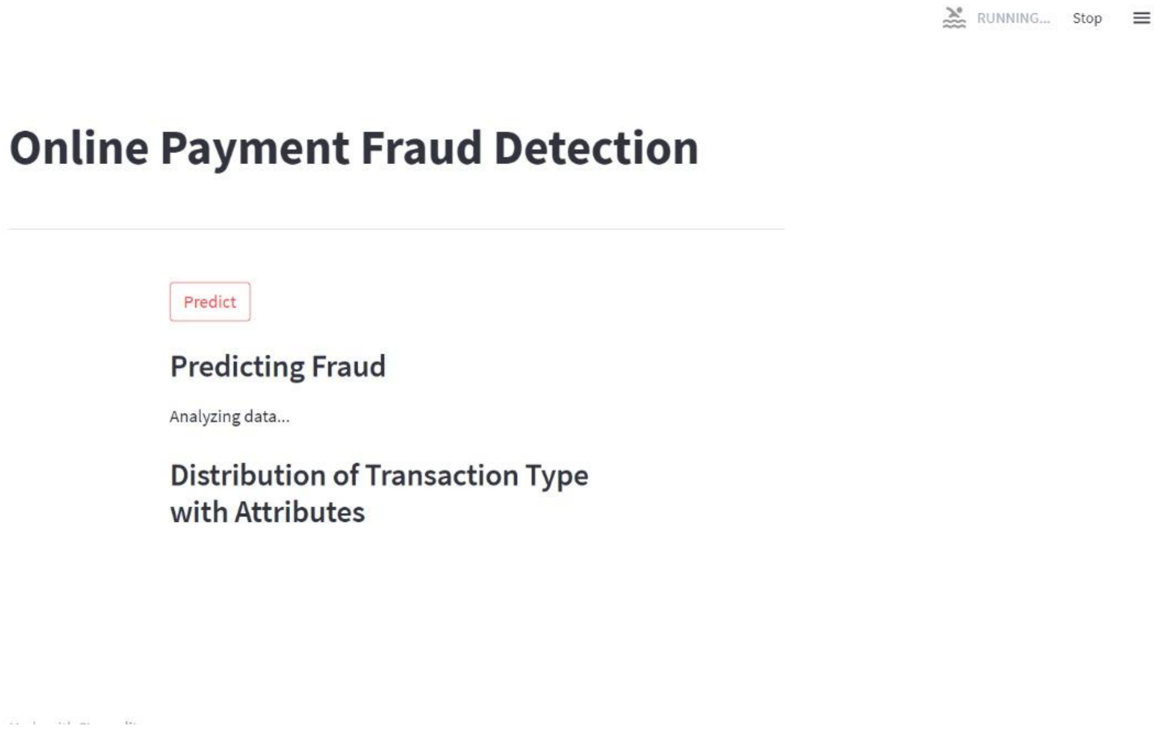
The ultimate goal of this project is to identify abnormalities in financial transactions by using the Decision Tree Classifier to the online payment fraud detection dataset. The performance indicators of the model provide a thorough assessment of its capabilities. With an accuracy score of 0.88, the model is good at producing accurate predictions and can effectively tell the difference between real and fraudulent transactions.

At 0.91, precision—a crucial indicator of the percentage of real positives among all positively recognised cases—is noted. This demonstrates the model's proficiency in detecting cases of online payment fraud with accuracy, reducing false positives, and enhancing system resilience.

On the other hand, the recall rate is 0.82, which indicates that the model is successful in identifying a significant proportion of real fraudulent instances. Measuring recall, or the percentage of real positive cases that the model accurately detected, highlights the system's capacity to detect a significant number of instances of online payment fraud.

Table 1. Result of the Evaluation metrics

Measurement metrics	Final Result obtained
Accuracy	0.99974224445470576
Precision	0.9997394094258494
Recall	0.9997422445470576
Loss	164



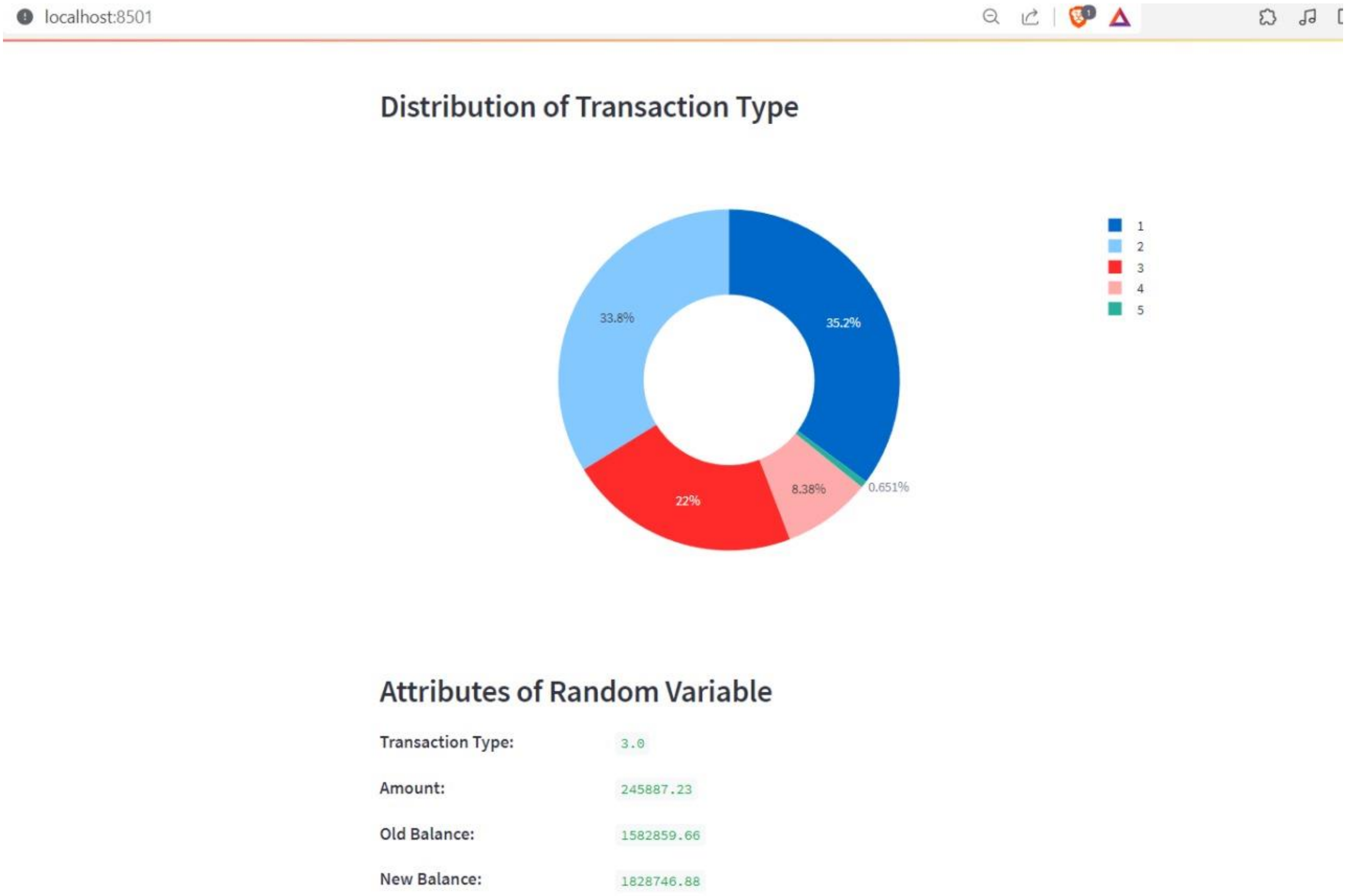
Online Payment Fraud Detection

Predicting Fraud

Analyzing data...

Distribution of Transaction Type with Attributes

Figure 7.2.1



Attributes of Random Variable

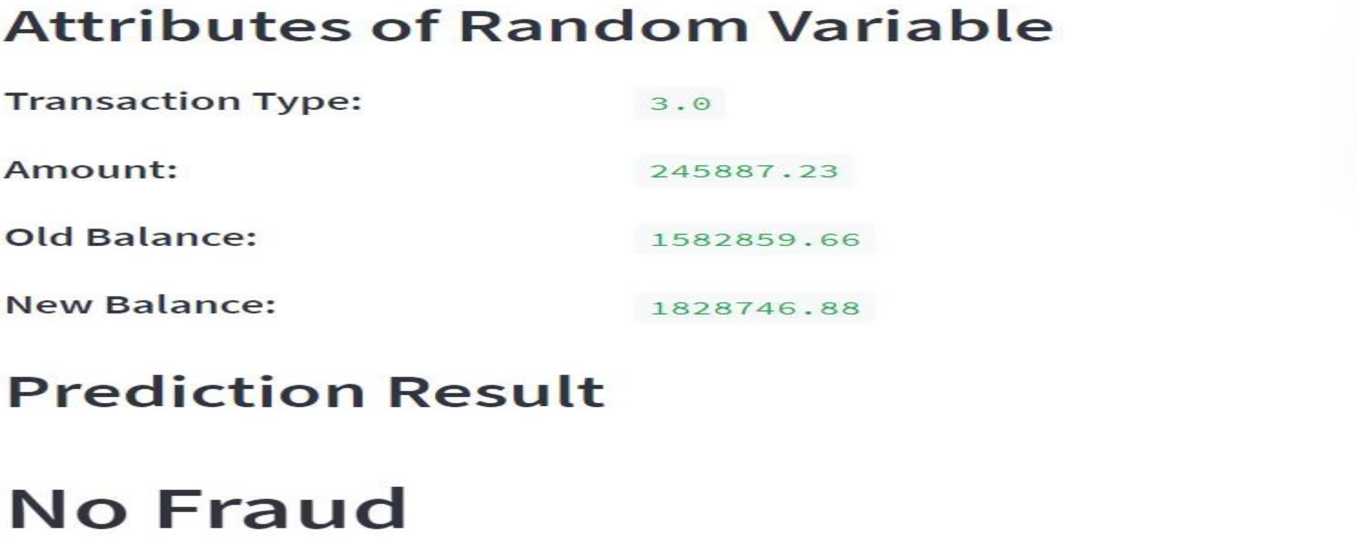
Transaction Type: 3.0

Amount: 245887.23

Old Balance: 1582859.66

New Balance: 1828746.88

Figure 7.2.2



Attributes of Random Variable

Transaction Type: 3.0

Amount: 245887.23

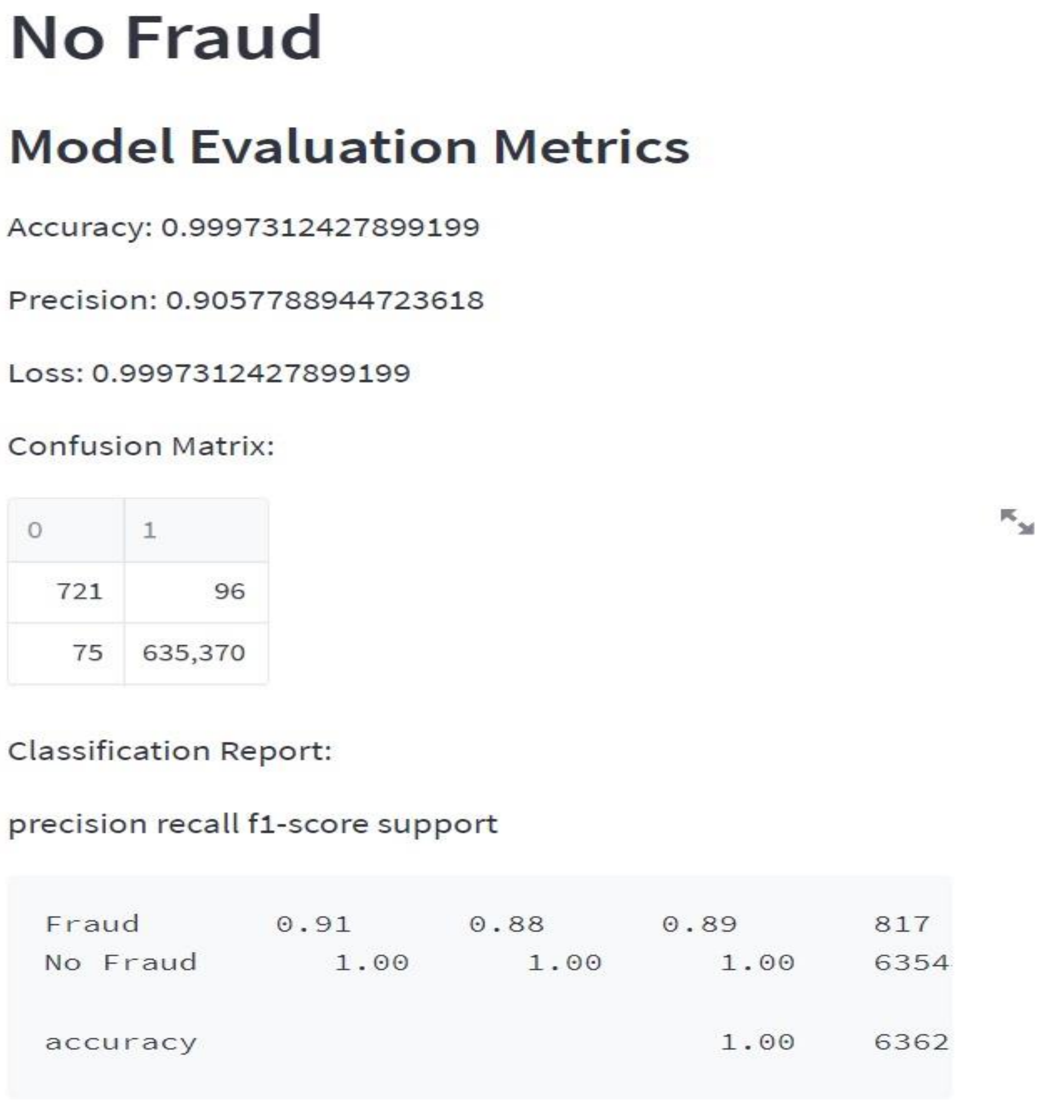
Old Balance: 1582859.66

New Balance: 1828746.88

Prediction Result

No Fraud

Figure 7.2.3



No Fraud

Model Evaluation Metrics

Accuracy: 0.9997312427899199

Precision: 0.9057788944723618

Loss: 0.9997312427899199

Confusion Matrix:

0	1
721	96
75	635,370

Classification Report:

precision recall f1-score support

```
Fraud         0.91      0.88      0.89      817
No Fraud         1.00      1.00      1.00    6354

accuracy                            1.00    6362
```

Figure 7.2.4

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION AND FUTURE ENHANCEMENT

The Outstanding results were obtained by using the Decision Tree Classifier to the online payment fraud detection dataset. The F1 Score of 86%, recall of 82%, precision of 91%, and high accuracy of 88% were achieved. When taken as a whole, these data confirm that the approach is effective in detecting and preventing online payment fraud. The increased accuracy highlights the model's ability to reduce false positives and avoid needless interference with valid transactions. Concurrently, the high recall suggests that the model may successfully identify a significant fraction of real fraudulent cases, which adds to the coverage's breadth. All things considered, the Decision Tree Classifier performs admirably, greatly boosting the security and dependability of digital financial transactions by offering a strong barrier against online payment fraud.

CHAPTER 9
REFERENCES

REFERENCES

[1] "Fraud Detection for Online Transactions Using Random Forest" by Jun Seok Kang, Hyunjung Shin, and Jongwon Kim. (2013)

[2] "Deep Learning Models for Online Fraud Detection" by M. H. Bhuyan, Dhruba Kumar Bhattacharyya, and J. K. Kalita. (2018)

[3] "A Study on Detection of Online Payment Frauds Based on Machine Learning Algorithms" by M. Nithya and R. Sumalatha. (2017)

[4] "Detecting Online Payment Fraud Using Machine Learning Techniques" by M. Akila, M. Hemalatha, and R. Ravi. (2019)

[5] "Fraud Detection in Online Payments Using Machine Learning" by R. Ravi, A. Sharmila, and K. Sujatha. (2016)

[6] Yu "Fraud Detection in Online Transactions Using Enhanced SVM" by Priyanka P. Bandagar and Ravindra K. Pardeshi. (2015)

[7] "Detecting Credit Card Fraud Using Neural Networks" by H. E. Bahrami, M. Esmaeili, and A. R. Movaghar. (2012)

[8] "Credit Card Fraud Detection Using Machine Learning: A Review" by S. R. Mangalwede, R. S. Kulkarni, and S. G. Kool. (2020)

[9] "A Survey on Credit Card Fraud Detection Methods" by T. Saranya, M. Sasikumar, and P. K. Sriman Narayanan. (2017)

[10] "Detecting Fraud in Online Transactions Using Support Vector Machine" by N. L. Sowmya and H. K. Srinivas. (2014)