FORECASTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPURAMU

In partial fulfilment of the requirements for the award of the degree
of

Bachelor of Technology

In

Computer Science and Engineering

By

M THAHER BASHA (21G31A0537)

Under the guidance of

Dr. P. VEERESH **M. Tech., Ph. D.**

Professor

Department. of Computer Science & Engineering

St. JOHNS COLLEGE OF ENGINEERING & TECHNOLOGY

Estd: 2001

Accredited by NAAC, Approved by AICTE, Recognized by UGC under 2(f) & 12(B), An ISO 9001:2015 Certified Institution and Affiliated to JNTUA, Anantapuramu

(AUTONOMOUS)

Yerrakota, YEMMIGANUR - 518360, Kurnool Dt., Andhra Pradesh.

Counseling Code: JONY

2021 – 2025

St. JOHNS COLLEGE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC, Approved by AICTE, Recognized by UGC under 2(f) & 12(B), An ISO 9001:2015 Certified Institution and Affiliated to JNTUA, Anantapuramu

(AUTONOMOUS)

Estd: 2001

Yerrakota, YEMMIGANUR - 518360, Kurnool Dt., Andhra Pradesh.

Counseling Code: JONY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled **—FORECASTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING** ‖ is bonafide work of **M Thaher Basha (21G31A0537)** submitted to the Department of Computer Science & Engineering, in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING from Jawaharlal Nehru Technological University, Anantapuramu.**

Signature of the Supervisor
Dr. P. VEERESH M. Tech., Ph. D.
Professor
St. Johns College of Engineering and Technology

Signature of the Head of the Dept.
Dr. P. VEERESH M. Tech., Ph. D.
H.O.D
St. Johns College of Engineering and Technology

Estd: 2001

St. JOHNS COLLEGE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC, Approved by AICTE, Recognized by UGC under 2(f) & 12(B), An ISO 9001:2015 Certified Institution and Affiliated to JNTUA, Anantapuramu

(AUTONOMOUS)

Yerrakota, YEMMIGANUR - 518360, Kurnool Dt., Andhra Pradesh.

Counseling Code: JONY

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

DECLARATION

I here by declare that the project Report entitled **—FORECASTING CREDIT CARD FRAUD DETECTION USING MACHINIE LEARNING** submitted by **M Thaher Basha (21G31A0537)** to the Department of Computer Science and Engineering, **St. Johns College of Engineering &Technology**, **Yerrakota, Yemmiganur, Kurnool,** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out by me under the supervision of Professor **Dr.P.VEERESH,** I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or any other institute or university.

Signature

M THAHER BASHA (21G31A0537)

St. JOHNS COLLEGE OF ENGINEERING & TECHNOLOGY

Accredited by NAAC, Approved by AICTE, Recognized by UGC under 2(f) & 12(B), An ISO 9001:2015 Certified Institution and Affiliated to JNTUA, Anantapuramu

(AUTONOMOUS)

Estd: 2001

Yerrakota, YEMMIGANUR - 518360, Kurnool Dt., Andhra Pradesh.

Counseling Code: JONY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

The project report entitled **―FORECASTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING ‖** is prepared and submitted by **M Thaher Basha (21G31A0537)** It has been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** in **St. Johns College of Engineering & Technology, Yerrakota, Yemmiganur, Kurnool, A.P.**

Guide

Dr.P.VEERESH
Professor Department of CSE

Head of the Department

Dr.P.VEERESH
Professor Department of CSE

Internal Examiner

External Examiner

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned my efforts with success. It is a pleasant aspect that I have now the opportunity to express my guidance for all of them.

The first and foremost, **Dr. P. VEERESH** **M. Tech., Ph. D.,** Professor of Computer Science and Engineering Department, who has extended her support for the success of this project. His wide knowledge and logical way of thinking have made a deep impression on me. His understanding, encouragement and personal guidance have provided the basis for this thesis. His source of inspiration for innovative ideas and his kind support is well to all his students and colleagues.

I express my thanks to Project Coordinator, **Mrs. S.S.RAJAKUMARI** **M.Tech,Ph.D**, for her Continuous support and encouragement.

I wish to thank **Dr. P. VEERESH** **M. Tech., Ph. D.,** Head of Computer Science and Engineering Department. His wide support, knowledge and enthusiastic encouragement have impressed me to better involvement into my project thesis and technical design also his ethical morals helped me to develop my personal and technical skills to deploy my project in success.

I wish to thank **Dr.K.Sudhakar**, Principal of St. Johns College of engineering and Technology who has extended his support for the success of this project.

I express my sincere thanks to the project committee members, faculty and staff of Computer Science and Engineering Department, St. Johns College of engineering and Technology, for their valuable guidance and technical support.

Last but far from least, I also thank my family members and my friends for their moral support and constant encouragement, I am are very much thankful to one and all who helped me for the successful completion of the project

With gratitude

M THAHER BASHA (21G31A0537)

CHAPTER CONTENTS PAGE NO

LIST OF FIGURES

ABSTRACT

Identification of bank cards fraud remains a serious worry for financial companies because of how sophisticated criminal behavior is becoming.. This study addresses this challenge by leveraging models increase the efficiency of detection systems. Utilizing a dataset from Kaggle, we implement and compare five distinct algorithms: LSTM networks, neural networks based on CNN, Decision Trees, Random Forests, and a Stacking Classifier. CNNs are employed to capture intricate patterns in transaction data, while LSTM address sequential dependencies and temporal dynamics. Decision Trees and Random Forests provide robust classification through hierarchical decision-making and ensemble learning. Additionally, a Stacking Classifier integrates the strengths of these algorithms to potentially improve overall performance. The comparative analysis of these methods tries to determine the best method for real-time identification of fraud. The results are expected to contribute significantly to the development of more secure credit card transaction systems, thereby mitigating financial losses and enhancing consumer trust.

CHAPTER 1: INTRODUCTION

1. INTRODUCTION

1.1 Overview

Credit card fraud detection is a vital aspect of the financial industry, designed to protect both institutions and consumers from financial losses. As fraudulent activities become more sophisticated, traditional detection methods are struggling to keep pace with the evolving tactics used by cybercriminals. Fraudulent practices like identity theft, account takeover, and transaction manipulation are now being carried out using advanced technologies and algorithms, making detection increasingly difficult.

Fraudsters employ complex techniques, including artificial intelligence, to bypass conventional detection systems, highlighting the need for more advanced, adaptable fraud detection mechanisms. These sophisticated threats demand real-time, dynamic detection systems to identify and combat fraud effectively.

The economic consequences of credit card fraud are severe, leading to substantial losses for financial institutions and damage to their reputations. Consumers are also impacted by financial losses and emotional distress. Consequently, there is a pressing need for more effective and advanced fraud detection systems that can evolve alongside these new and complex threats.

1.2 Motivation

Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both institutions and consumers against financial losses. One major concern is the growing sophistication of fraudulent activity. Necessitating advanced detection methods to combat evolving threats. Credit card fraud encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation. These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics employed by cybercriminals.

The trends in fraudulent activities are continually evolving, with attackers leveraging increasingly complex methods to evade traditional detection systems. For instance, fraudsters now use advanced algorithms and artificial intelligence to generate and execute fraudulent transactions, making it harder for conventional systems to identify anomalies effectively. As a result, there is a growing need for more sophisticated fraud detection systems that can adapt to these new threats in real-time.

The economic impact of credit card fraud is substantial. Financial institutions face significant losses due to fraudulent transactions, which can also damage their reputation and consumer trust. Consumers, on the other hand, may suffer financial losses, emotional distress, and inconvenience due to fraud. The direct financial implications are often accompanied by indirect costs, such as increased security measures and regulatory compliance. Addressing these challenges through advanced fraud detection systems is crucial in order to reduce monetary losses, mizing financial losses, enhancing consumer confidence, as well as preserving the banking ecosystem's trustworthinessing the integrity of the financial system. Thus, improving fraud detection capabilities is not only a technical not only a need but also a crucial strategic move for the business of finance.

1.3 Problem Defination

Fraud detection remains A critical concern for financial organazations, as traditional methods struggle to keep pace with the rapid evolution of fraudulent tactics. Conventional methods for detecting fraud frequently depend on rule-based techniques, which involve predefined criteria and patterns for identifying suspicious activities. While these systems can be effective against known types of fraud, they are inherently limited by their static nature. They often fail to detect new, sophisticated fraud schemes that do not match the established rules or patterns. This limitation leaves significant gaps in fraud prevention, allowing innovative fraudulent activities to go undetected. Another significant challenge is the dynamic Fraudsters are constantly evolving their techniques and strategies to carry out fraudulent activities. They consistently modify and refine their methods to stay ahead of detection systems. exploit vulnerabilities in detection systems. They employ advanced techniques such as social engineering, synthetic identities, and sophisticated phishing schemes, which traditional systems may not adequately address. This adaptability creates a constant arms race between fraudsters and detection systems, with fraudsters frequently outpacing the defenses designed to protect against them.

Additionally, traditional systems often rely on limited features or simplistic models that may not capture the complex and multifaceted nature of fraud. They may also suffer from high rates of Untrue positives which raise operating expenses and customer dissatisfaction. As fraud tactics evolve, More sophisticated and adaptable detecting technologies are desperately needed methods. Machine intelligence offers a promising solution, providing the capability to examine enormous volumes of transaction data and identify subtle, previously undetected patterns indicative of fraudulent activity. Thus, enhancing fraud detection systems with machine learning techniques could bridge the gap left by traditional methods and improve overall detection accuracy and efficiency.

1.4 Objective

Main goal of this investigation is to leverage cutting-edge ML techniques to enhance the accuracy and efficiency of detection systems. Fraud remains a major challenge. For financial institutions because of their increasing sophistication of fraudulent activities. Traditional techniques for detecting cheating often fall short in addressing these evolving threats, necessitating the exploration of innovative solutions.

To achieve this, the study focuses on applying and evaluating four distinct machine intelligence algorithms: CNN, LSTM , Decision Trees, and Random Forests. Each algorithm offers unique strengths that can contribute to the detection of fraudulent transactions.

- **CNN:** are employed to capture intricate patterns and anomalies within transaction data. CNNs excel in identifying complex features through their hierarchical layers, making them suitable for detecting subtle signs of fraud.

- **LSTM** : are utilized to address sequential dependencies and temporal dynamics inherent in transaction data. LSTM are particularly effective in analyzing time-series data, which is crucial for detecting fraud patterns that evolve over time.

- **Decision Trees**: provide a transparent, hierarchical approach to classification, making decisions based on feature values. They offer interpretability and are useful for understanding decision rules.

- **Random Forests:** build upon Decision Trees by employing an ensemble learning approach. This method aggregates using choice trees to enhance classification accuracy. Accuracy and reduce overfitting, enhancing the overall robustness of the fraud detection system.

This study seeks to determine the most efficient method for real-time fraud detection by evaluating and comparing various algorithms. The anticipated outcomes should provide significant insights into enhancing fraud prevention strategies. security and efficiency of credit card transaction systems, thereby mitigating financial losses and enhancing consumer trust.

1.5 Limitations of the Project

This study employs a Kaggle dataset to Create and assess ML models for the detection of credit card fraud using the dataset provided. a critical component of this research, comprises transaction records with labeled instances of fraudulent and legitimate activities. The dataset encompasses multiple attributes, including the transaction amount and timestamp. and user-specific details, which are essential for training and testing the algorithms. However, the dataset may come with inherent limitations, including potential imbalances between fraudulent and legitimate transactions. This class imbalance can affect the performance of the algorithms, leading to challenges in achieving high accuracy and minimizing false positives and negatives.

In terms of scope, the study focuses on four distinct machine intelligence algorithms: CNN, LSTM, Decision Trees, and Random Forests. Each algorithm is selected to explore different aspects of fraud detection—CNNs for capturing intricate patterns, LSTM for understanding sequential dependencies, and Decision Trees and Random Forests for robust classification. The aim is to compare these methods to determine the most successful strategy for real-time fraud detection.

Despite these efforts, several limitations must be acknowledged. Information integrity issues, like absent or inconsistent data, can negatively affect the precision of the models., the generalizability of the findings may be constrained by the dataset's specificity and the potential overfitting of models to the training data. Algorithmic constraints, such as computational complexity and model interpretability, may also affect the practical implementation of the proposed solutions in real-world systems. Addressing these limitations will be crucial for enhancing Durability and effectiveness of systems for detecting fraud.

1.6 ORGANIZATION OF THE PROJECT

In addressing the critical issue of credit card fraud detection, this study seeks to answer several key research questions that are central to Improving the precision and effectiveness of fraud detection systems through AI.

- **What are the advantages and disadvantages of every algorithm for detecting fraud?** This question focuses on understanding how CNN, LSTM, Decision Trees, and Random Forests perform in detecting fraudulent transactions. Each algorithm offers unique advantages and limitations. CNNs are recreated for their strength to grasp Complex designs and features within transaction data, making them effective in identifying subtle anomalies. LSTM are particularly adept at managing and interpreting sequential and temporal dependencies, which is crucial for detecting fraud in time-series data. Decision Trees provide a clear hierarchical decision-making process, while Random Forests, with their ensemble approach, enhance classification robustness by reducing overfitting and improving generalization. This comparative analysis will help to elucidate the specific scenarios where each algorithm excels or falls short.

- **How can machine learning improve real-time fraud detection?** This question explores the potential of ML to advance real-time fraud detection capabilities. Traditional methods often struggle with the dynamic nature of fraudulent activities, which requires adaptive and rapid response mechanisms. Subset of AI algorithms, with their potential to take in and make use of large amounts of data and adapt to new patterns, offer a significant advantage. By leveraging these algorithms monetary establishments can potentially improve their capacity to finding and react to fraudulent activities instantly, thus minimizing financial damages. and increasing consumer trust. This study aims to provide insights into how artificial intelligence may be applied successfully utilized to address the challenges of real-time fraud detection

This is to follow up the next chapters i.e., Chapter 2 contains the information about the system specifications. It clearly explains the libraries offered by the system. Software requirements and hardware requirements are also mentioned in the chapter. The next chapter i.e., Chapter 3 deals with the design and implementation of the project. It covers the technology that is used for the project. It also contains the source code of the project and the output screenshots of the project. The last chapter i.e., Chapter 4 provides the concluding information of the project. The report ends with a list of the references that have been used

CHAPTER 2 : LITERATURE SURVEY

2 LITERATURE SURVEY

2.1 INTRODUCTION

Related work :

1. R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao (2022)

Title: Credit Card Fraud Detection Using Machine Learning

This study compares two popular machine learning algorithms, Random Forest and Adaboost, for credit card fraud detection. The paper evaluates both algorithms based on several metrics, including accuracy, precision, recall, F1-score, and the ROC curve. Through this comparison, the authors aim to determine which method is the most effective for detecting fraud in credit card transactions.

2. A. Thennakoon, C. Bhagvani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarac (2019)

Title: Real-time Credit Card Fraud Detection Using Machine Learning

This paper explores the use of machine learning models for real-time credit card fraud detection. The authors evaluate optimal algorithms for detecting various types of fraud and implement a system for real-time fraud detection through predictive analytics. An API is also used to facilitate the real-time application of the system, providing efficient fraud detection solutions.

3 1. J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare (2017)

Title: Credit card fraud detection using machine learning techniques: A comparative analysis

In this paper, the authors highlight the critical role of data mining in detecting credit card fraud. The study emphasizes the challenges posed by adaptive detection mechanisms and handling imbalanced datasets, particularly when working with real-time transaction data. Through a comparative analysis of various machine learning techniques, the paper offers insights into how data mining can enhance the accuracy and efficiency of fraud detection systems.

4. F. K. Alarfaj, L. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Aluned (2022)

Title: Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms

This paper introduces a hybrid system combining an autoencoder with a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) for credit card fraud detection. The proposed model aims to address the challenges of detecting financial fraud with high accuracy, particularly in the context of imbalanced datasets. The authors use oversampling techniques to balance the dataset, which significantly improves both recall and precision in fraud detection, making it a highly effective solution in real-world applications.

5 D. Varmedia, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla (2019)

Title: Credit Card Fraud Detection Using Machine Learning Methods

In this research, the authors employ various machine learning techniques to detect credit card fraud. They use SMOTE (Synthetic Minority Over-sampling Technique) for balancing the dataset, which helps in dealing with the common problem of imbalanced classes in fraud detection tasks. The paper also applies feature selection to enhance the performance of machine learning models, including Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron (MLP). Their approach demonstrates high accuracy in detecting fraud despite the challenges of working with imbalanced datasets.

2.2 EXISTING SYSTEM

Current credit card fraud detection systems primarily rely on rule-based methods, statistical models, and traditional machine learning algorithms such as Logistic Regression and Support Vector Machines (SVM). These systems often face limitations in detecting complex and evolving fraud patterns due to their reliance on static rules and the inability to capture temporal dependencies in transaction data. As fraudsters develop more sophisticated techniques, these systems struggle to maintain high accuracy and timely detection, leading to false positives and false negatives, which can result in either customer dissatisfaction or financial losses for institutions

2.3 Disadvantages of Existing System

- **Static Rules:** Rule-based systems depend on predefined rules that may not adapt to new or evolving fraud patterns. As fraudsters develop more sophisticated techniques, these static rules can quickly become outdated, leading to missed detections.

- **Limited Flexibility:** Traditional statistical models and algorithms like Logistic Regression and SVM are often rigid and may not effectively capture the complexities of transaction data, limiting their ability to identify nuanced fraudulent behaviors.

- **Inability to Capture Temporal Dependencies:** Many traditional methods do not account for the temporal dynamics of transaction data, making it difficult to recognize patterns that evolve over time. This oversight can result in missed detections of fraud that occurs in a sequence.

- **High False Positive Rates:** Relying on static rules can lead to a high number of false positives, where legitimate transactions are incorrectly flagged as fraudulent. This can frustrate customers and damage their trust in the financial institution.

- **High False Negative Rates:** Conversely, these systems may also fail to detect actual fraudulent transactions, leading to financial losses for institutions and increased risk for customers.

- **Limited Adaptability:** As fraud tactics evolve, traditional systems often require manual updates to rules and models, which can be time-consuming and may not keep pace with the speed of fraud development.

2.4 PROPOSED SYSTEM

The proposed method for enhancing credit card fraud detection involves the implementation and comparison of five machine learning algorithms: Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), Decision Trees, Random Forests, and a Stacking Classifier. Each algorithm is chosen for its unique strengths in handling complex transactional data. CNNs are utilized to identify intricate patterns within the data, while LSTMs are employed to capture sequential dependencies and temporal relationships. Decision Trees offer straightforward, interpretable decision-making, and Random Forests enhance this by aggregating multiple trees to improve classification accuracy. The Stacking Classifier integrates predictions from the CNN, LSTM, Decision Trees, and Random Forests to create a more robust and comprehensive model. This method aims to optimize detection accuracy and efficiency, thereby reducing false positives and improving real-time detection capabilities in credit card fraud prevention systems.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The proposed method offers several advantages, including improved detection accuracy by combining multiple algorithms, the ability to capture both spatial and temporal transaction patterns, and the robustness provided by ensemble learning. This approach enhances the system's capability to detect sophisticated fraud schemes, reduces false positives, and increases the reliability of fraud detection, ultimately contributing to better protection against financial losses and higher consumer trust in transaction systems.

CHAPTER 3: SYSTEM SPECIFICATIONS

3 .SYSTEM SPECIFICATION

3.1 SOFTWARE SPECIFICATIONS

Operating System	: Windows 7/8/10
Server side Script	: HTML, CSS, Bootstrap & JS
Programming Language	: Python
Libraries	: Flask, Pandas, Mysql.connector, Os, Tensorflow, Keras Numpy
IDE/Workbench	: Visual Code
Technology	: Python 3.10
Server Deployment	: Xampp Server
Database	: MySQL

3.2 HARDWARE SPECIFICATIONS

Processor	- I3/Intel Processor
Hard Disk	- 160GB
Key Board	- Standard Windows Keyboard
Mouse	- Two or Three Button Mouse
Monitor	– SVGA
RAM	- 8GB

3.3 NON FUNCTIONAL REQUIREMENTS

Nonfunctional requirements are the functions offered by the system. It includes time constraints on the development process and standards. The non-functional requirements are as follows:

Speed: The system should process the given input into output within an appropriate time.

Ease of use: The software should be user-friendly. Then the customers can use it easily, so it does not require much training time.

Reliability: The rate of failure should be less then only the system is more reliable

Portability: It should be easy to implement in any system.

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
2) The processing of each request should be done within 10 seconds
3) The site should load in 3 seconds whenever of simultaneous users are > 10000

CHAPTER 4 : FORECASTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

4. PREDICTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

4.1 MACHINE LEARNING BASED FORECASTING CREDIT CARD

DATA DESCRIPTION

- 1Unnamed: 0 ==> This feature represents Likely an index or row identifier from the dataset. This column might not hold meaningful data and can often be dropped.

- 2trans_date_trans_time ==> This Componet indicates The timestamp of the transaction, typically in a date-time notation (e.g., HH:MM:SS or YYYY-MM-DD).

- 3cc_num ==> This feature indicates The credit card number used for the transaction. This is sensitive information and should be handled with care

- 4.merchant ==> This aspect indicates The name of the merchant or store where the transaction occurred.

- 5.category ==> This attribute records the kind or category of products or services that were bought (e.g., groceries, electronics).
- 6.amt ==> it's represents The amount of money spent in the transaction.

- 7. first ==> This section indicates The first name of the cardholder.

- 8. last ==> This feature captures The last name of the cardholder.

- 9. gender ==> it shows The gender of the cardholder (e.g., Male, Female).

- 10. street ==> The place of residence connected to the hyping information of the cardholder.

- 11.city ==> This aspect captures The merchant billing address city.

- 12.state ==> This feature indicates The status of the advertising address.

- 13.zip ==> This col represents The postal ZIP code associated with the cardholder's address.

- 14.lat ==> This component indicates The azimuth of the merchant's latitude coordinate.

- 15.long ==> This feature represents The geographic location reflected in the merchant's longitude coordinate.

- 16.city_pop ==> This variable represents The number of people residing in the city where the transaction took place.
- 17.job ==> This Columns indicates This feature representsThe occupation of the cardholder.

- 18.dob ==> it tells Record the cardholder's date of birth.

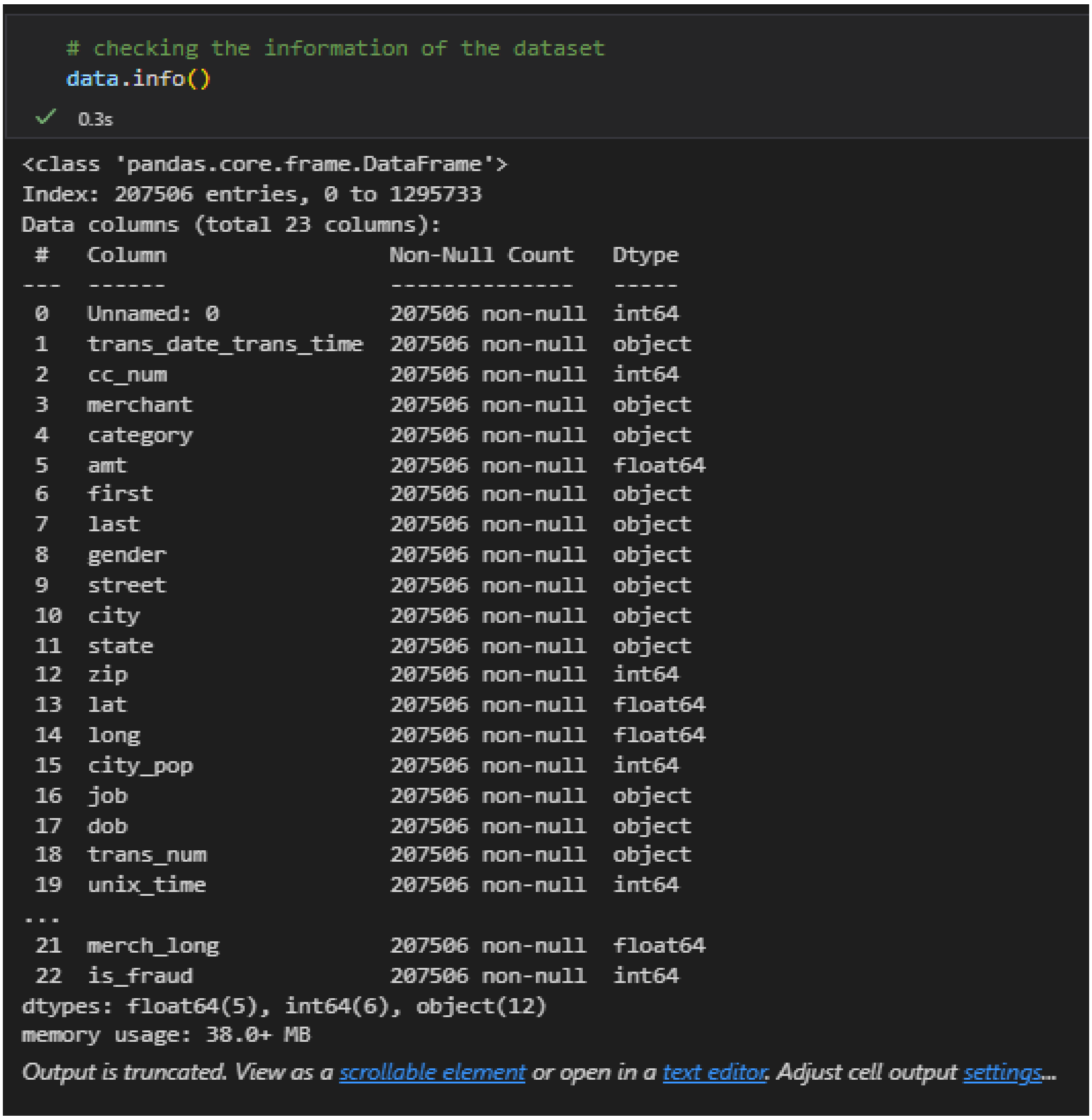
- 19. trans_num ==> This feature donates The unique transaction number or ID.

- 20.unix_time ==> it represents The timestamp of the transaction in Unix epoch time format.

- 21.merch_lat ==> This columns shows the latitude organize of where the merchant is located

- 22.merch_long ==> Column indicates the longitude coordinate of the holder location.

- 23.is_fraud ==> its represents A binary indicator (usually 0 or 1) showing whether the transaction was fraudulent (1) or not (0).



```
# checking the information of the dataset
data.info()
```

✓ 0.3s

```
<class 'pandas.core.frame.DataFrame'>
Index: 207506 entries, 0 to 1295733
Data columns (total 23 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   Unnamed: 0             207506 non-null  int64
 1   trans_date_trans_time  207506 non-null  object
 2   cc_num                 207506 non-null  int64
 3   merchant               207506 non-null  object
 4   category               207506 non-null  object
 5   amt                    207506 non-null  float64
 6   first                  207506 non-null  object
 7   last                   207506 non-null  object
 8   gender                 207506 non-null  object
 9   street                 207506 non-null  object
 10  city                   207506 non-null  object
 11  state                  207506 non-null  object
 12  zip                    207506 non-null  int64
 13  lat                    207506 non-null  float64
 14  long                   207506 non-null  float64
 15  city_pop               207506 non-null  int64
 16  job                    207506 non-null  object
 17  dob                    207506 non-null  object
 18  trans_num              207506 non-null  object
 19  unix_time              207506 non-null  int64
...
 21  merch_long             207506 non-null  float64
 22  is_fraud               207506 non-null  int64
dtypes: float64(5), int64(6), object(12)
memory usage: 38.0+ MB
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Data Collection

In this study, we utilize a dataset sourced from Kaggle, which is widely recognized for its extensive collection of data suitable for machine intelligence applications. The dataset is designed specifically for fraud detection and includes a diverse range of features essential for identifying fraudulent transactions. It comprises transaction details such as transaction amount, timestamp, and anonymized variables derived from credit card transactions, which capture various aspects of user behavior and transaction patterns.

The dataset features a substantial number of transactions, with both legitimate and fraudulent entries It is necessitate to assure that the dataset is well-balanced to avoid bias in the training

process. In this case, the dataset contains a mix of both types, although fraudulent transactions are significantly less frequent, reflecting real-world scenarios where frauds are comparatively uncommon in contrast to reputable deals.

To prepare the dataset for machine learning models, several preprocessing steps are undertaken. Data cleaning is first carried out to address missing values and eliminate any outliers that might distort the outcomes. Data normalization is applied to normalize the range of features. Which helps in improving the performance of Implement machine learning algorithms while ensuring that every feature has an equal impact on the model's learning process..

Feature engineering is also a vital process that entails generating new features or modifying existing ones to boost the model's performance. to detect fraud. Finally, the dataset is divided into test, validation, and training sets to assess the performance of the machine intelligence models effectively. This structured approach ensures The information is utilized for training and testing is representative of real-world conditions, providing a solid foundation for building and assessing fraud detection systems.

Data Preprocessing

Effective data training is crucial for building robust subset of AI models, especially in the context of identificaton fraud where data quality can significantly impact model performance. The preprocessing stage involves several key steps:

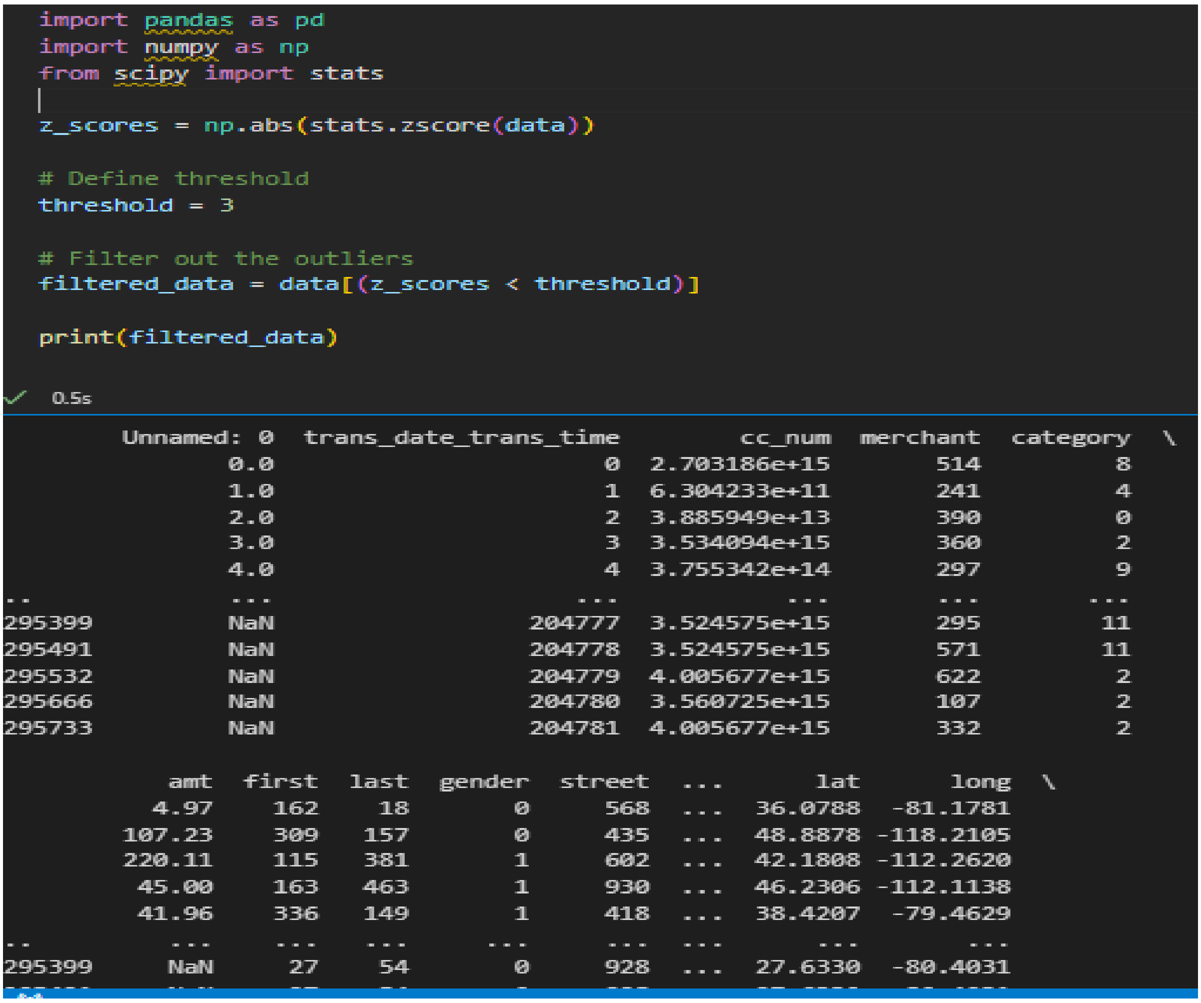
Handling Missing Values:

Null values in the dataset are Located to ensure that the machine learning algorithms operate on complete data. Common methods include inference, in which statistical measurements are used to replace missing valuesSuch as the standard deviation, the median, or mode of operation, or more complex approaches. like k-Nearest Neighbors imputation. Alternatively, rows or columns with excessive missing values may be removed if they do not provide critical information.

Normalization:

To guarantee that different features contribute evenly to the model's performance., normalization is applied. This involves scaling feature values to a standard range, typically between 0 and 1 or -1 and 1. Methods such

as Z-score Uniformity and Min-Max Scaling are commonly used. Normalization helps in stabilizing the training process and improving the convergence rate of algorithms like The networks made up of Long Short-Term Memory LSTM and Convolutional neural networks, more commonly are sensitive to the size of input features.

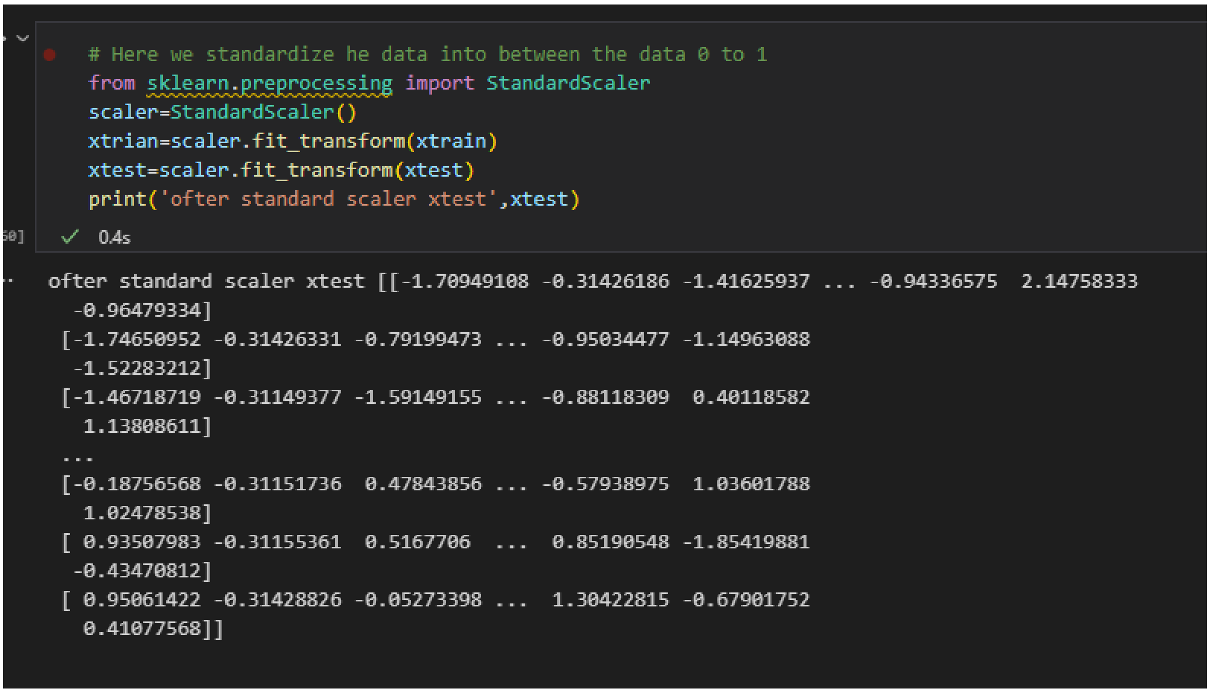


```
import pandas as pd
import numpy as np
from scipy import stats

z_scores = np.abs(stats.zscore(data))

# Define threshold
threshold = 3

# Filter out the outliers
filtered_data = data[(z_scores < threshold)]

print(filtered_data)
```

✓ 0.5s

```
        Unnamed: 0  trans_date_trans_time        cc_num  merchant  category  \
               0.0                      0  2.703186e+15       514         8
               1.0                      1  6.304233e+11       241         4
               2.0                      2  3.885949e+13       390         0
               3.0                      3  3.534094e+15       360         2
               4.0                      4  3.755342e+14       297         9
...            ...                    ...           ...       ...       ...
295399         NaN                 204777  3.524575e+15       295        11
295491         NaN                 204778  3.524575e+15       571        11
295532         NaN                 204779  4.005677e+15       622         2
295666         NaN                 204780  3.560725e+15       107         2
295733         NaN                 204781  4.005677e+15       332         2

           amt  first  last  gender  street  ...      lat      long  \
          4.97    162    18       0     568  ...  36.0788  -81.1781
        107.23    309   157       0     435  ...  48.8878 -118.2105
        220.11    115   381       1     602  ...  42.1808 -112.2620
         45.00    163   463       1     930  ...  46.2306 -112.1138
         41.96    336   149       1     418  ...  38.4207  -79.4629
...        ...    ...   ...     ...     ...  ...      ...       ...
295399     NaN     27    54       0     928  ...  27.6330  -80.4031
```

Figure 1using Z- score for outliers



```
# Here we standardize he data into between the data 0 to 1
from sklearn.preprocessing import StandardScaler
scaler=StandardScaler()
xtrian=scaler.fit_transform(xtrain)
xtest=scaler.fit_transform(xtest)
print('ofter standard scaler xtest',xtest)
```

✓ 0.4s

```
ofter standard scaler xtest [[-1.70949108 -0.31426186 -1.41625937 ... -0.94336575  2.14758333
  -0.96479334]
 [-1.74650952 -0.31426331 -0.79199473 ... -0.95034477 -1.14963088
  -1.52283212]
 [-1.46718719 -0.31149377 -1.59149155 ... -0.88118309  0.40118582
   1.13808611]
 ...
 [-0.18756568 -0.31151736  0.47843856 ... -0.57938975  1.03601788
   1.02478538]
 [ 0.93507983 -0.31155361  0.5167706  ...  0.85190548 -1.85419881
  -0.43470812]
 [ 0.95061422 -0.31428826 -0.05273398 ...  1.30422815 -0.67901752
   0.41077568]]
```

Figure 2 using Standardion

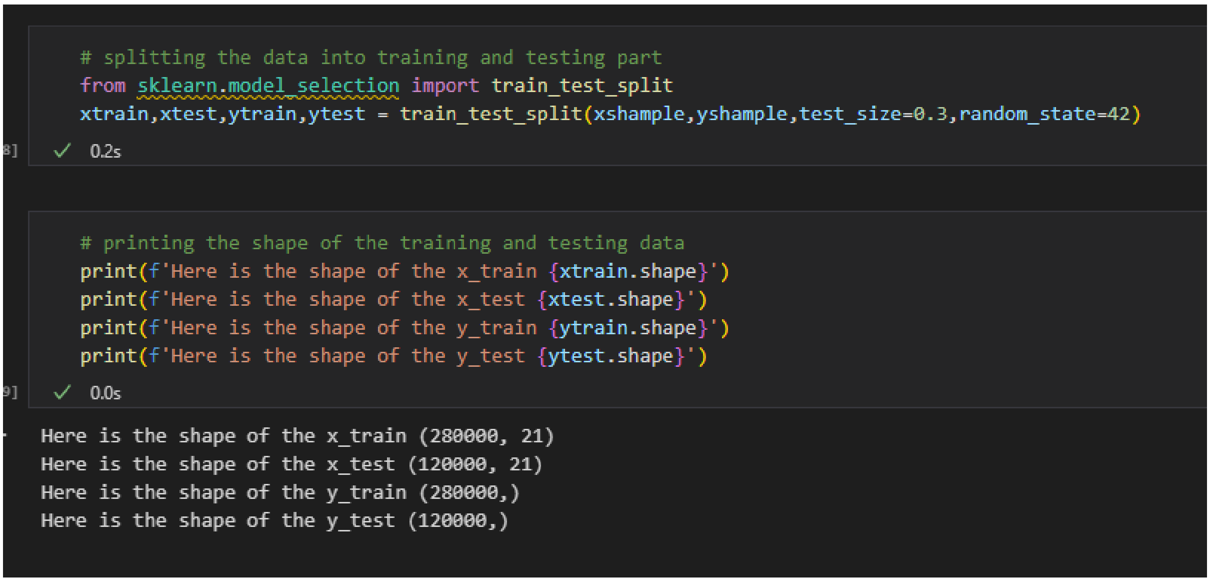
Feature Selection:

Selecting relevant features is essential to enhance model efficiency and accuracy. Feature selection methods, such as statistical tests, correlation analysis, and techniques like Recurring Structures Feature Elimination, or RFE or Principal Component Analysis (PCA), are used to identify and retain the most informative features while discarding redundant or irrelevant ones. This step helps in reducing dimensionality and mitigating the risk of overfitting.

Data Splitting:

The dataset is separated into training, validation, and test sets to evaluate model performance effectively. Typically, the data is split into Ten to fifteen percent for testing, seventy to eighty percent for training. The test set is used to gauge how well the final model can generalize, the validation set is used to adjust hyper parameters and monitor performance throughout training, and the training set is used to train the models.

These preprocessing procedures make sure the data is representative, clean, and scalable, which improves the efficiency of subset of AI algorithms. in detecting fraudulent transactions accurately.



```
# splitting the data into training and testing part
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(xshample,yshample,test_size=0.3,random_state=42)
```

✓ 0.2s

```
# printing the shape of the training and testing data
print(f'Here is the shape of the x_train {xtrain.shape}')
print(f'Here is the shape of the x_test {xtest.shape}')
print(f'Here is the shape of the y_train {ytrain.shape}')
print(f'Here is the shape of the y_test {ytest.shape}')
```

✓ 0.0s

```
Here is the shape of the x_train (280000, 21)
Here is the shape of the x_test (120000, 21)
Here is the shape of the y_train (280000,)
Here is the shape of the y_test (120000,)
```

Figure 3 data splitting

Algorithm Implementation

The implementation of algorithms for credit Detecting card theft requires a methodical approach to implementing CNN, LSTM, Decision Trees, and Random Forests. Each algorithm is tailored to address specific aspects of fraud detection and enhance the overall system's accuracy and efficiency.

4.2. MODULE DESCRIPTION

1.CNN:

The CNN architecture is structured to achieve and detect complex patterns in transaction data. The implementation begins with defining the network's layers, including feature extraction using convolutional layers that pooling layers for dimensionality reduction, and fully connected layers for classification. The convolutional layers use filters to scan through the transaction data, identifying intricate patterns that might indicate fraudulent activity. Pooling layers help in reducing the computational load by summarizing the features extracted. In order to minimize the loss function, the network is trained using gradient descent and backpropagation. and optimize performance.

2. LSTM:

LSTM networks are configured to handle sequential data, capturing temporal dependencies crucial for detecting fraudulent patterns over time. The implementation involves setting up LSTM cells that maintain and update internal states through forget, input, and output gates. This structure allows the LSTM to remember long-term dependencies and recognize patterns in transaction sequences. Training involves using sequences of transaction data, with the LSTM adjusting its parameters to learn the temporal relationships between events.

3. Stacking Classifier

A Stacking Classifier is an ensemble learning method that blends several machine learning models rise predictive performance. Unlike traditional ensemble methods like bagging or boosting, which use a single type of model, stacking integrates different types of models to leverage their unique strengths.

In a stacking classifier, the process is typically divided into two levels. At the first level, various base models are trained on the same dataset. Each base model independently predicts the output. These predictions, along with the original features, are then passed as the data source for a second-level model, also known as a meta-classifier or meta-learner.

4. Decision Trees and Random Forests:

To create decision forests, split data at each node to classify transactions, and build a hierarchical model that makes judgments based on feature values. By training an ensemble of decision trees using a randomized a portion of the characteristics and data, Random Forests expand on this concept. The forest aggregates The different trees' forecasts show increased robustness. and accuracy. Both algorithms are trained and evaluated using metrics such as accuracy, precision, and recall, to ensure effective fraud detection.

5. LIME Implementation

LIME (Local Interpretable Model-agnostic Explanations) is a technique that explains predictions of machine learning models. It works by approximating a complex model with a simpler, interpretable model locally around the prediction. LIME perturbs input data, observes the changes in predictions, and uses this information to fit an interpretable model (like linear regression) that approximates the behavior of the complex model for that specific instance. This provides insight into the model’s decision-making process.

Model Training and Tuning

In this study, the model training process is designed to optimize the performance of five ML algorithms specifically for fraud detection. Each model undergoes extensive training using a preprocessed dataset from Kaggle, ensuring that the data is properly balanced to resolve the class disparity that exists in fraud detection activities..

For the CNN model, a grid search is conducted to fine-tune hyper parameters such as the number of convolutional layers, filter sizes, and learning rates. The LSTM model, given its ability to capture temporal dependencies, is optimized by adjusting the number of LSTM units, dropout rates, and sequence lengths. Decision Trees are pruned and tuned for depth and split criteria to avoid overfitting. Random Forests, being an ensemble of decision trees, are tuned by varying To find a Establish a balance between variation and bias, then calculate the maximum depth with all tree.

The Stacking Classifier combines the forecasts made by each separate model to generate a more reliable forecast. The base models for stacking are trained with optimal hyper parameters obtained from prior tuning processes. A meta-learner, typically a logistic regression or another tree-based model, is then trained on the outputs of these base models.

Cross-validation techniques are employed throughout the tuning process to validate the models' generalization performance, ensuring that the final models are not only accurate but also capable of handling real-world credit card transaction data efficiently.

Optimization techniques such as grid search or random search are employed to systematically explore different hyper parameter combinations and identify the optimal settings. Cross-validation is used to evaluate model performance and ensure that it generalizes well to unseen data. This comprehensive training and tuning process target to boost the overall performance of the fraud detection system, resulting in a more precise and trustworthy identification of fraudulent transactions.

Evaluation Metrics

In preforming subset of AI models for fraud identifications, several key metrics are utilized to assess performance comprehensively. These metrics—precision, recall, accuracy, and the F1-score—all shed light on various facets of the model's efficacy. in detecting fraudulent transactions.

Accuracy

Evaluates the model's broad precision by multiplying the quantity of correctly anticipated occurrences by the total number of instances.. While accuracy provides a general sense of performance, it can be deceptive when there is an imbalance datasets where fraudulent transactions are much less frequent than legitimate ones.

Precision

is crucial for assessing the quality of positive predictions, described as the proportion of accurate favorable forecasts to the overall amount of the model's successful predictions. High accuracy denotes78 that when the model predicts fraud, it is likely to be correct, thus minimizing false positives.

Recall

Also known as sensitivity, it measures how effectively the model is able to locate all relevant

examples of fraud. It is finding as the Percentage of true instance to the total number of actual positives (no fraud transactions). High recall ensures that most fraudulent transactions are detected, although it may come at the cost of lower precision.

F1-score

Integrates recall and precision into one statistic, offering a balanced view of model performance. It is the The harmonic definition of recall and precision, offering a measure that considers both false positives and false negatives, making it particularly useful in scenarios with class imbalance.

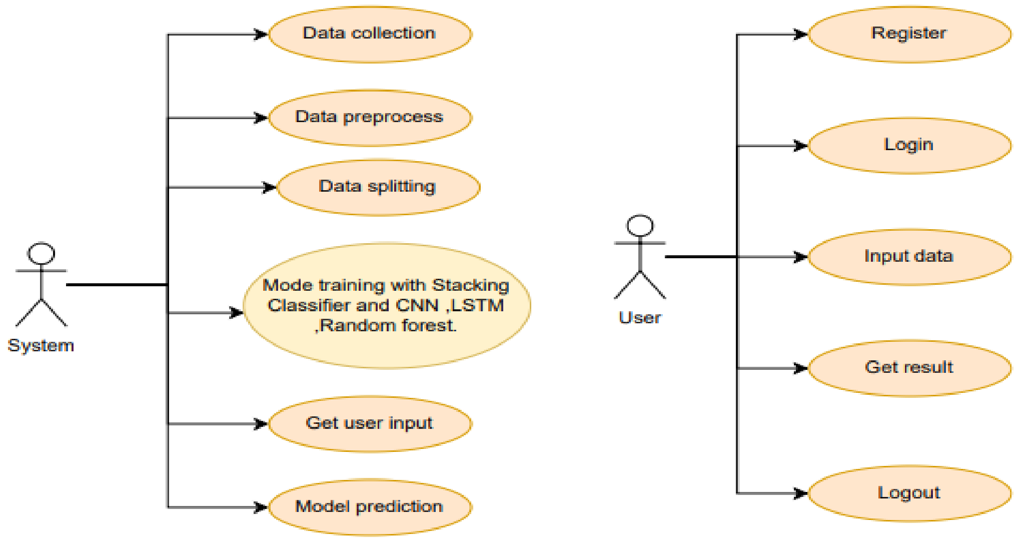
Cross-validation

The frameworks are tested for resilience using techniques such as k-fold validated cross-valid The dataset has been separated into k subsets using this manner, and training and testing the model on these k-1 subsets and confirming its results on the remaining subset. Every subset serves as the validation set once during the k iterations of this process. This procedure aids in evaluation To what extent the model lowers the chance of overfitting and generalizes to new data.

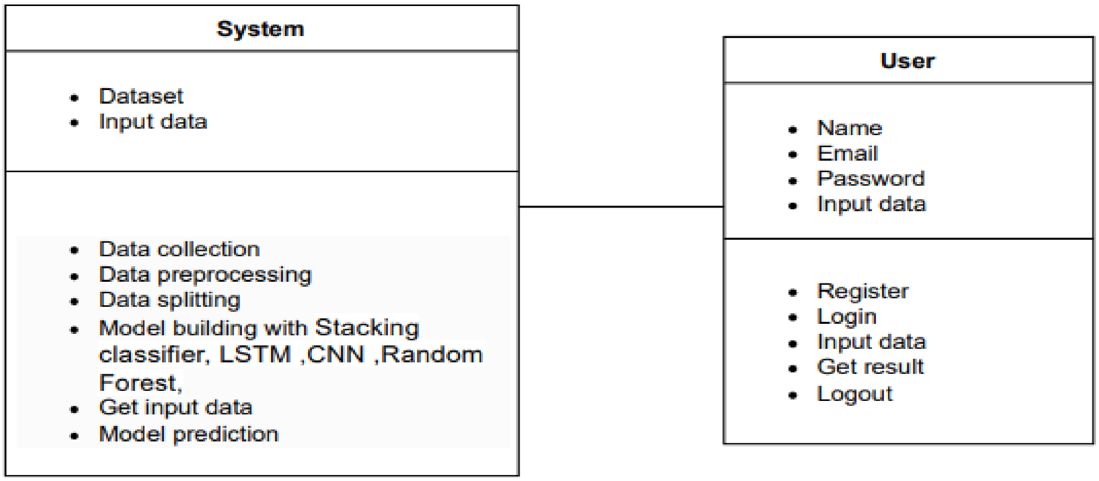
Together, these evaluation measures and techniques ensure a thorough assessment of model performance, helping to identify the most effective algorithm for real-time credit card fraud detection.

4.3 UML DIAGRAMS

USE CASE DIAGRAM



Class Diagram:



BLOCK DIAGRAM

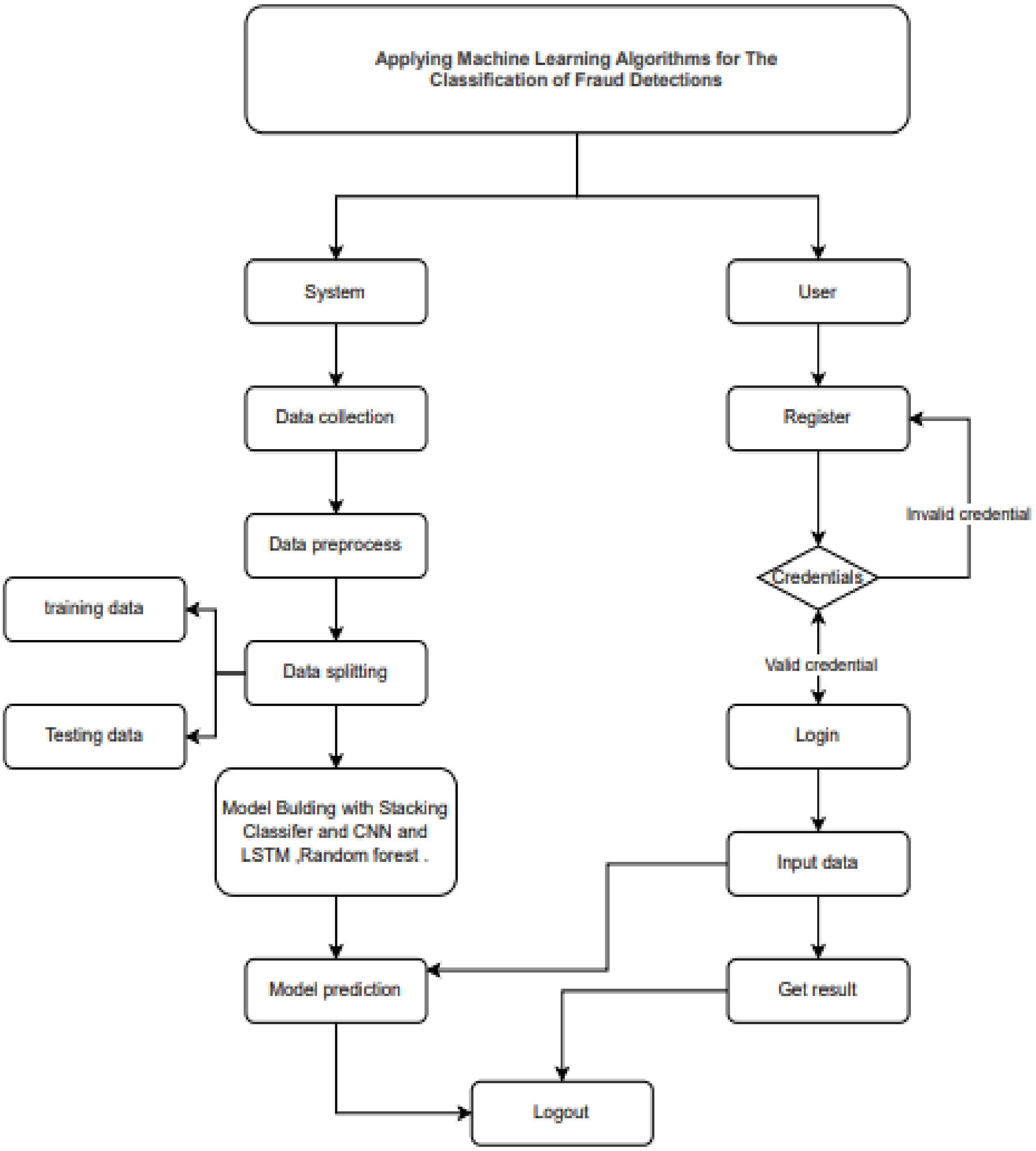


Figure 4 block diagram

Architecture Diagram

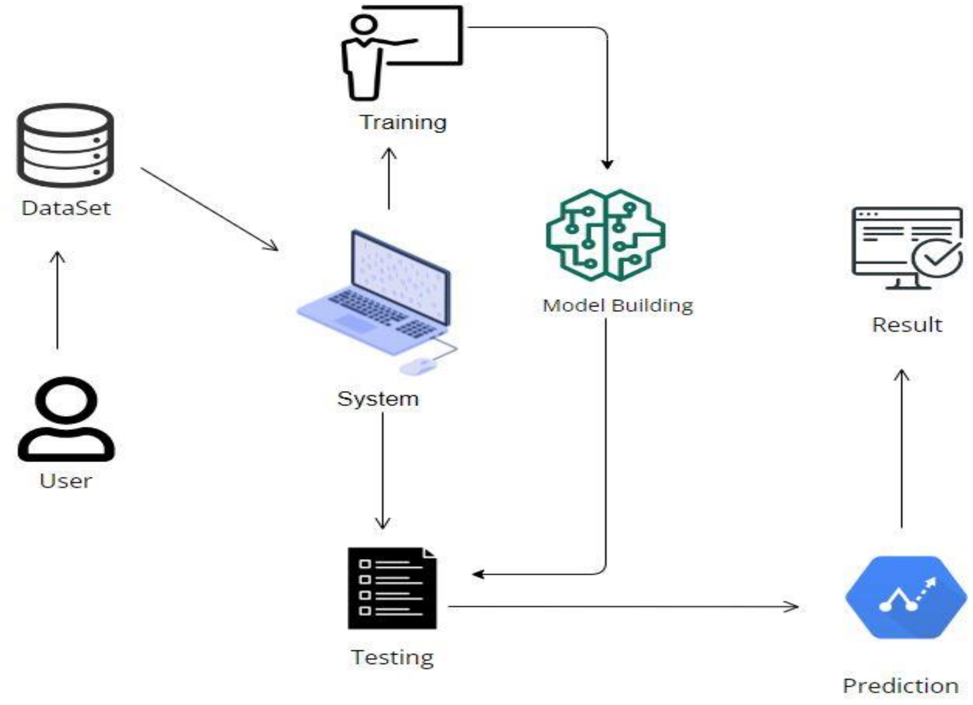
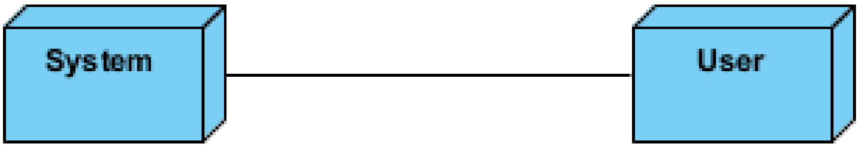
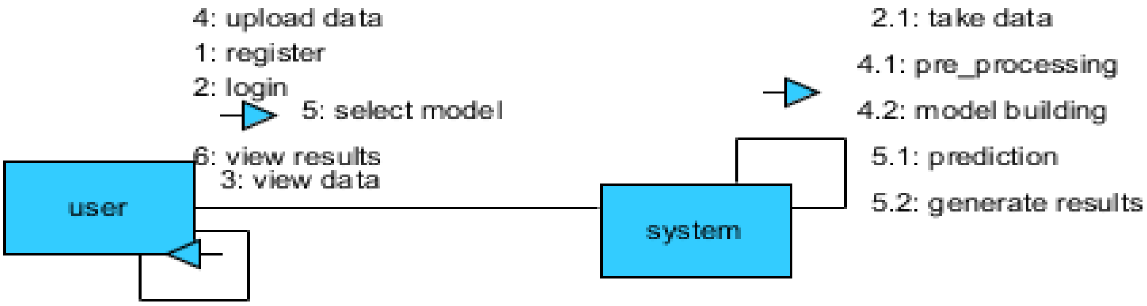


Figure 5 architecture diagram

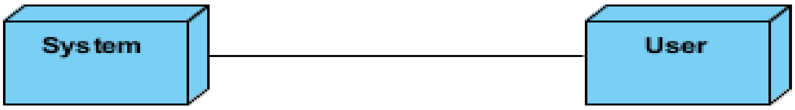
SEQUENCE DIAGRAM



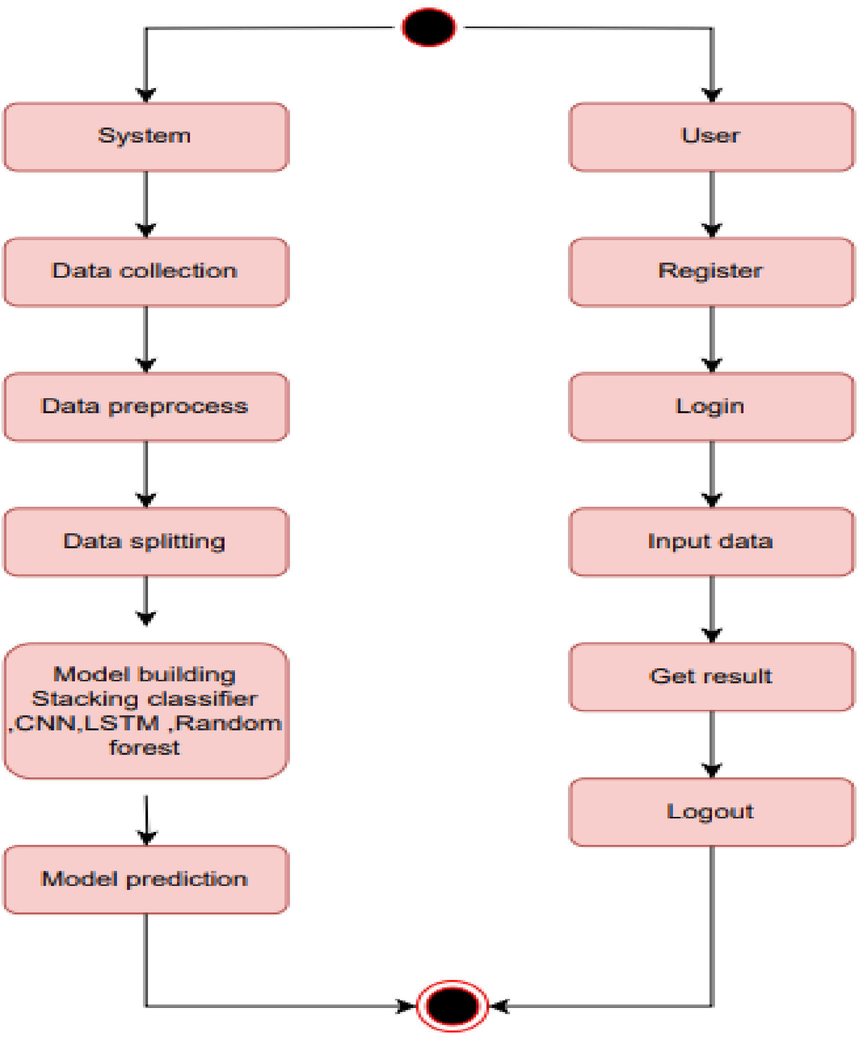
Collaboration Diagram:



Deployment Diagram

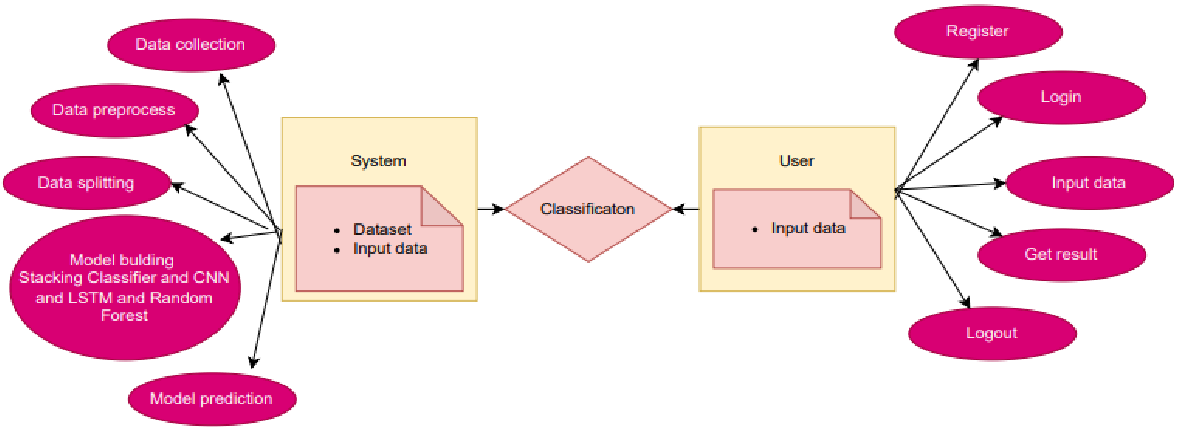


Activity Diagram:

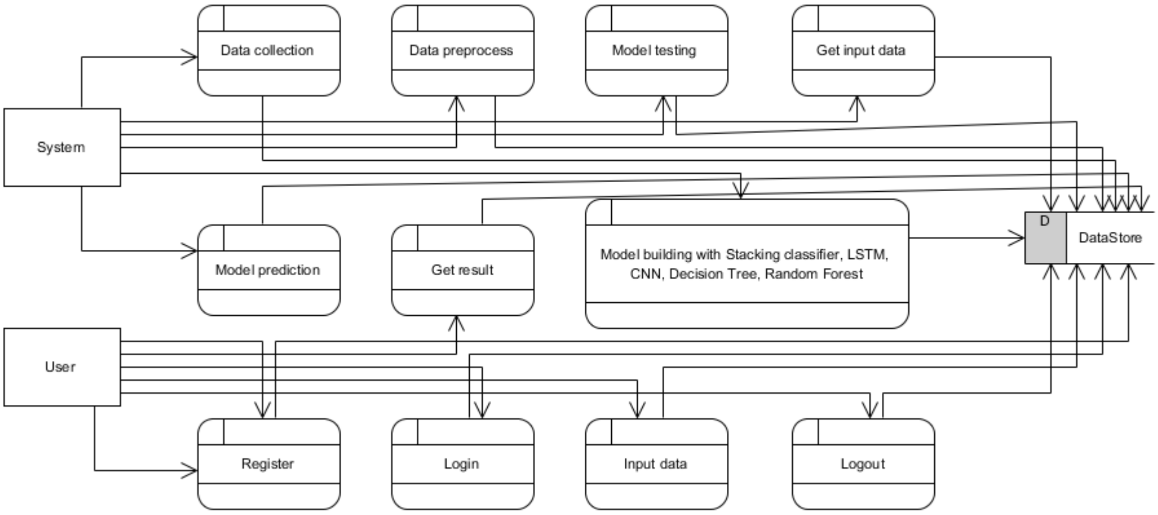


ER Diagram:

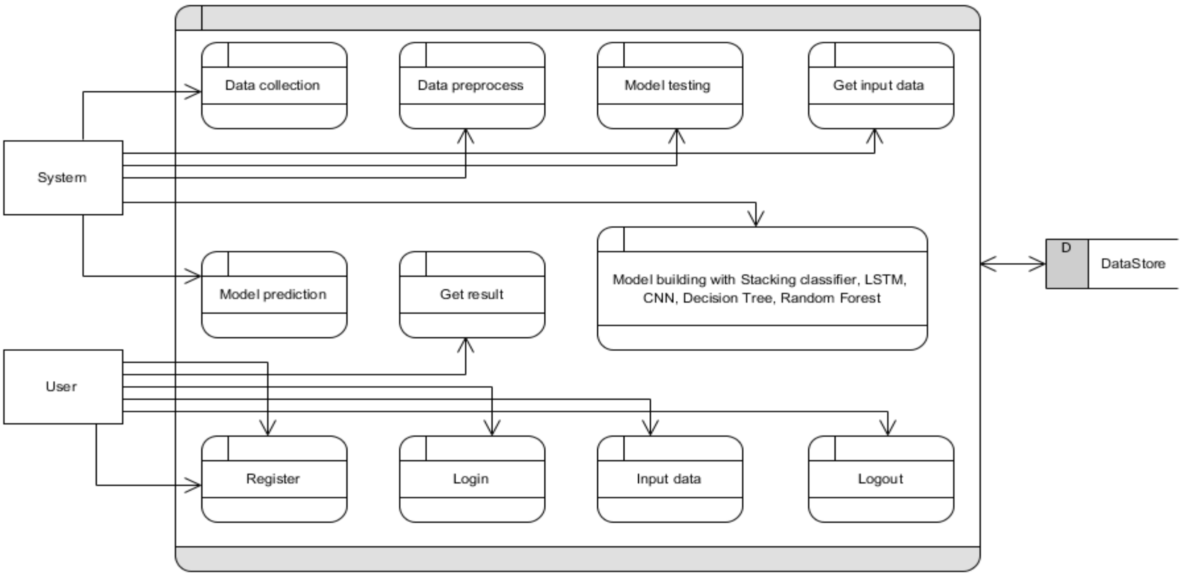
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.


Data collection
Data preprocess
Data splitting
Model bulding
Stacking Classifier and CNN
and LSTM and Random
Forest
Model prediction
System
Dataset
Input data
Classificaton
User
Input data
Register
Login
Input data
Get result
Logout

DFD Diagram:


Data collection
Data preprocess
Model testing
Get input data
System
Model prediction
Get result
Model building with Stacking classifier, LSTM,
CNN, Decision Tree, Random Forest
D
DataStore
User
Register
Login
Input data
Logout

DFD Diagram: Level 2:



4.4 SOURCE CODE

4.4.1 FRONT-END CODE

#About.html

```
{% extends 'index.html' %}


{% block navbar %}
  <li><a href="{{url_for('index')}}">Home</a></li>
  <li class="active"><a href="{{url_for('about')}}">About</a></li>
  <li><a href="{{url_for('register')}}">Register</a></li>
  <li><a href="{{url_for('login')}}">Login</a></li>
{% endblock %}


{% block content %}
  <section class="section why-us" data-section="section2">
    <div class="container" style="height: 600px;">
      <div class="row">
        <div class="col-md-12">
          <div id='tabs'>
            <section class='tabs-content' style="margin-top: 80px;">
              <article id='tabs-1'>
                <div class="row">
                  <div class="col-md-6">
                    <img src="/static/images/concerned-man-with-devices-medium-shot.jpg" height="420px" alt=""
style="margin-top: 140px;">
                  </div>
                  <div class="col-md-6">
                    <h4>About</h4>
                    <p>
                      Credit card fraud detection is a critical component of the financial sector, essential for
safeguarding both institutions and consumers against financial losses. The increasing sophistication of  fraudulent
activities poses a significant challenge, necessitating advanced detection methods to combat evolving threats. Credit
card fraud encompasses a range of deceptive practices, including identity theft, account takeover, and transaction
manipulation. These fraudulent activities have become  more sophisticated due to the advent of new technologies and
```

tactics employed by cybercriminals. The trends in fraudulent activities are continually evolving, with attackers leveraging increasingly complex methods to evade traditional detection systems.

For instance, fraudsters now use advanced algorithms and artificial intelligence to generate and execute fraudulent transactions, making it harder for conventional systems to identify anomalies effectively. As a result, there is a growing need for more sophisticated fraud detection systems that can adapt to these new threats in real-time.

```
</p>
                  </div>
                </div>
              </article>
            </section>
          </div>
        </div>
      </div>
    </div>
  </section>
{% endblock %}
```

#CONTACT.HTML

```
<!DOCTYPE html>
<html>

<head>
  <!-- Basic -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- Mobile Metas -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <!-- Site Metas -->
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <title>Neogym</title>
  <!-- slider stylesheet -->
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css?family=Poppins:400,600,700&display=swap" rel="stylesheet">
  <!-- Custom styles for this template -->
  <link href="css/style.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="css/responsive.css" rel="stylesheet" />
</head>

<body class="sub_page">
  <div class="hero_area">
    <!-- header section strats -->
    <header class="header_section">
      <div class="container-fluid">
        <nav class="navbar navbar-expand-lg custom_nav-container ">
          <a class="navbar-brand" href="index.html">
            <span>
              Neogym
            </span>
          </a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>

          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <div class="d-flex ml-auto flex-column flex-lg-row align-items-center">
              <ul class="navbar-nav  ">
                <li class="nav-item ">
                  <a class="nav-link" href="index.html">Home <span class="sr-only">(current)</span></a>
                </li>
                <li class="nav-item ">
        <a class="nav-link" href="why.html"> Why us </a>
       </li>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="trainer.html"> trainers</a>
       </li>
       <li class="nav-item active">
        <a class="nav-link" href="contact.html"> Contact Us</a>
       </li>
      </ul>
      <div class="user_option">
       <form class="form-inline my-2 my-lg-0 ml-0 ml-lg-4 mb-3 mb-lg-0">
        <button class="btn  my-2 my-sm-0 nav_search-btn" type="submit"></button>
       </form>
      </div>
     </div>
    </div>
   </nav>
  </div>
 </header>
 <!-- end header section -->

</div>



<!-- contact section -->

<section class="contact_section ">
 <div class="container-fluid">
  <div class="row">
   <div class="col-md-6 px-0">
    <div class="img-box">
     <img src="images/contact-img.jpg" alt="">
    </div>
   </div>
   <div class="col-lg-5 col-md-6">
    <div class="form_container pr-0 pr-lg-5 mr-0 mr-lg-2">
     <div class="heading_container">
```

```
          <h2>
            Contact Us
          </h2>
        </div>
        <form action="">
          <div>
            <input type="text" placeholder="Name" />
          </div>
          <div>
            <input type="email" placeholder="Email" />
          </div>
          <div>
            <input type="text" placeholder="Phone Number" />
          </div>
          <div>
            <input type="text" class="message-box" placeholder="Message" />
          </div>
          <div class="d-flex ">
            <button>
              Send
            </button>
          </div>
        </form>
      </div>
    </div>
  </div>
 </div>
</section>

<!-- end contact section -->

<!-- info section -->
<section class="info_section layout_padding2">
 <div class="container">
  <div class="info_items">
   <a href="">
    <div class="item ">
     <div class="img-box box-1">
```

```
          <img src="" alt="">
        </div>
        <div class="detail-box">
          <p>
            Location
          </p>
        </div>
      </div>
    </a>
    <a href="">
      <div class="item ">
        <div class="img-box box-2">
          <img src="" alt="">
        </div>
        <div class="detail-box">
          <p>
            +02 1234567890
          </p>
        </div>
      </div>
    </a>
    <a href="">
      <div class="item ">
        <div class="img-box box-3">
          <img src="" alt="">
        </div>
        <div class="detail-box">
          <p>
            demo@gmail.com
          </p>
        </div>
      </div>
    </a>
  </div>
</div>
</section>

<!-- end info_section -->
```

```
  <!-- footer section -->
  <footer class="container-fluid footer_section">
    <p>
      &copy; 2020 All Rights Reserved. Design by
      <a href="https://html.design/">Free Html Templates</a>
    </p>
  </footer>
  <!-- footer section -->

  <script src="js/jquery-3.4.1.min.js"></script>
  <script src="js/bootstrap.js"></script>

</body>

</html>
```

# DATABASE.HTML

```
DROP DATABASE IF EXISTS db;
CREATE DATABASE db;
USE db;

CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    email VARCHAR(50),
    password VARCHAR(50)
);
```

# HOME.HTML

```
<!DOCTYPE html>
<html>

<head>
```

```
  <!-- Basic -->
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <!-- Mobile Metas -->
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
  <!-- Site Metas -->
  <meta name="keywords" content="" />
  <meta name="description" content="" />
  <meta name="author" content="" />

  <title>CREDIT CARD</title>

  <!-- slider stylesheet -->
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="/static/css/bootstrap.css" />

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css?family=Poppins:400,600,700&display=swap" rel="stylesheet">
  <!-- Custom styles for this template -->
  <link href="/static/css/style.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="/static/css/responsive.css" rel="stylesheet" />
</head>

<body>
  <div class="hero_area">
    <!-- header section strats -->
    <header class="header_section">

      <div class="container-fluid">
        <nav class="navbar navbar-expand-lg custom_nav-container ">
          <!-- <a class="navbar-brand" href="Home.html"> -->
            <span style="color: rgb(237, 244, 250); font-size: 25px;">
              <u style="color: rgb(255, 254, 252);">
              PREDICTING CREDIT CARD FRAUD DETECTION
            </u>
            </span>
          </a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <div class="d-flex ml-auto flex-column flex-lg-row align-items-center">

              <ul class="navbar-nav  ">
                <li class="nav-item ">
                  <a class="nav-link" href="{{url_for('Home')}}">Home</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{url_for('model')}}">model</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{url_for('Prediction')}}">Prediction</a>
                </li>
                <li class="nav-item">
                  <a class="nav-link" href="{{url_for('index')}}">Logout</a>
                </li>
                </ul>
            </div>
          </div>
        </nav>
      </div>
    </header>

{% block content %}
    <!-- end header section -->
    <!-- slider section -->
    <section class=" slider_section position-relative">
      <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
        <div class="carousel-inner">

          <div class="carousel-item active">
        <div class="container">
          <div class="col-lg-10 col-md-11 mx-auto">
            <div class="detail-box">
              <div>
                <h5>
                  PREDICTING
                </h3>
                <h4>
                  CREDIT CARD FRAUD DETECTION
                </h2>
                <h2>
                  USING MACHINE LEARNING
                </h2>
                <pb style="font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
font: size 20px;">
                  Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both
institutions and consumers against financial losses. The increasing sophistication of fraudulent activities poses a
significant challenge, necessitating advanced detection methods to combat evolving threats.Credit card fraud
encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation.
These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics
employed by cybercriminals.</p>
              </div>
            </div>
          </div>
        </div>
      </div>
     {% endblock %}
   <script src="js/jquery-3.4.1.min.js"></script>
   <script src="js/bootstrap.js"></script>

</body>

</html>
```

# Index.HTML

```
<!DOCTYPE html>
<html lang="en">
```

```
  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,700,800,900"
rel="stylesheet">

    <title>Credit card detection</title>

    <!-- Bootstrap core CSS -->
    <link href="/static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="/static/css/fontawesome.css">
    <link rel="stylesheet" href="/static/css/templatemo-grad-school.css">
    <link rel="stylesheet" href="/static/css/owl.css">
    <link rel="stylesheet" href="/static/css/lightbox.css">

    {% block extra_style %}
    {% endblock %}
  </head>

<body>



  <header class="main-header clearfix" role="header">
    <div class="logo">

<a href="#" style="font-size: 25px;">credit <em style="color: rgba(201, 228, 49, 0.87);">Frud detection</em></a>
    </div>
    <nav id="menu" class="main-nav" role="navigation">
      <ul class="main-menu">
        {% block navbar %}
          <li class="active"><a href="{{url_for('index')}}">Home</a></li>
          <!--  -->
        <li><a href="{{url_for('about')}} ">About</a></li>
        <!-- -->
        <li><a href="{{url_for('register')}}">Register</a></li>
        <!--  -->
        <li><a href="{{url_for('login')}}">Login</a></li>
        <!--  -->
       {% endblock %}
     </ul>
   </nav>
  </header>

  {% block content %}
   <section class="section main-banner" id="top" data-section="section1">

       <img src="/static/images/pexels-tima-miroshnichenko-6266632.jpg" height="100%" width="100%" style="margin-
top: -200px;">

       <div class="video-overlay header-text">
         <div class="caption">
           <h4 style="margin-top: -50px; color: rgb(202, 218, 62);">WELCOME!</h4>
           <h2 style="font-size: 50px; color: rgba(229, 247, 128, 0.37);">Applying Machine Learning Algorithms for the
<em style="color:rgba(201, 228, 49, 0.87) ;">Credit card fraud detection</em></h2>
           <div class="main-button">
             <div class="scroll-to-section" style="display: inline-block; margin-right: 10px; ">
               <a href="{{url_for('register')}}" style="width: 200px; color: rgb(7, 7, 5);background-color: rgba(201, 228, 49,
0.87);">Register</a>
             </div>
             <div class="scroll-to-section" style="display: inline-block;">
               <a href="{{url_for('login')}}" style="width: 200px; color: black ;background-color: rgba(201, 228, 49,
0.87);">Login</a>
             </div>
           </div>
         </div>
       </div>
   </section>
  {% endblock %}

  {% if message %}
   <script>
     alert("{{message}}")
   </script>
  {% endif %}

  {% block extra_script %}
  {% endblock %}

  <!-- Scripts -->
  <!-- Bootstrap core JavaScript -->
   <script src="/static/vendor/jquery/jquery.min.js"></script>
   <script src="/static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

   <script src="/static/js/isotope.min.js"></script>
   <script src="/static/js/owl-carousel.js"></script>
   <script src="/static/js/lightbox.js"></script>
   <script src="/static/js/tabs.js"></script>
   <script src="/static/js/video.js"></script>
   <script src="/static/js/slick-slider.js"></script>
   <script src="/static/js/custom.js"></script>
   <script>
       //according to loftblog tut
       $('.nav li:first').addClass('active');

       var showSection = function showSection(section, isAnimate) {
        var
        direction = section.replace(/#/, ''),
        reqSection = $('.section').filter('[data-section="' + direction + '"]'),
        reqSectionPos = reqSection.offset().top - 0;

        if (isAnimate) {
         $('body, html').animate({
           scrollTop: reqSectionPos },
         800);
        } else {
         $('body, html').scrollTop(reqSectionPos);
        }
       };

       var checkSection = function checkSection() {
        $('.section').each(function () {
         var
         $this = $(this),
         topEdge = $this.offset().top - 80,
         bottomEdge = topEdge + $this.height(),
         wScroll = $(window).scrollTop();
         if (topEdge < wScroll && bottomEdge > wScroll) {
           var
           currentId = $this.data('section'),
           reqLink = $('a').filter('[href*=\\#' + currentId + ']');
           reqLink.closest('li').addClass('active').
           siblings().removeClass('active');
         }
        });
       };

       // $('.main-menu, .scroll-to-section').on('click', 'a', function (e) {
       //   if($(e.target).hasClass('external')) {
       //     return;
       //   }
       //   e.preventDefault();
       //   $('#menu').removeClass('active');
       //   showSection($(this).attr('href'), true);
       // });

       $(window).scroll(function () {
        checkSection();
       });
   </script>
 </body>
 </html>
```

# LOGIN.HTML

```
{% extends 'index.html' %}
{% block navbar %}
    <li><a href="{{url_for('index')}}">Home</a></li>
    <li><a href="{{url_for('about')}}">About</a></li>
    <li><a href="{{url_for('register')}}">Register</a></li>
    <li class="active"><a href="{{url_for('login')}}">Login</a></li>
{% endblock %}

{% block content %}
    <center>
        <section class="section coming-soon" data-section="section3" style="height: 700px;">
            <!-- <img src="/static/images/trainer-bg.jpg" height="100%" width="100%" alt="" style="background-position: 
center; background-size: cover;"> -->
            <div class="col-5" style="margin-top: 50px;">
                <form id="contact" action="{{url_for('login')}}" method="post">
                    <center>
                        <h1 style="color: white;">Login</h1><br>
                    </center>
                    <div class="row">

                        <div class="col-md-12">
                            <fieldset>
                                <input name="email" type="email" class="form-control" id="email" placeholder="Email ID" 
required="">
                            </fieldset>
                        </div>

                        <div class="col-md-12">
                            <fieldset>
                                <input name="password" type="password" class="form-control" id="password" 
placeholder="Password" required="">
                            </fieldset>
                        </div>

                        <div class="col-md-12">
                            <fieldset>
                                <button type="submit" id="form-submit" class="button" style="color: black;">Submit</button>
                            </fieldset>
                        </div>
                </div>
              </form>
            </div>
          </section>
        </center>
{% endblock %}
```

# MODEL.HTML

```
{% extends 'Home.html' %}

{% block content %}

<body style="background-color: #f0f0f0; font-family: sans-serif; ">

<div style="width: 500px; margin: 170px auto; background-color: #00000060; padding: 20px; border-radius: 5px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);">
  <h1 style="text-align: center; color: #f3eaea;">Model Training</h1>



  {% if msg %}
  <p style="color: rgb(206, 201, 157); text-align: center;">{{ msg }}</p>
  {% endif %}

  <form action="{{url_for('model')}}" method="post"enctype="multipart/form-data">
    <label for="algorithm" style="display: block; margin-bottom: 5px; color: #aca2a2;">Select an Algorithm:</label>
    <select id="algorithm" name="algorithm" style="width: 100%; padding: 10px; border: 1px solid #a08888; border-radius: 3px; box-sizing: border-box;">
      <option>Choose Algorithm</option>
      <option value="1">Stacking Classifier</option>
      <option value="2">Random_Forest Classifier</option>
      <option value="3">CNN</option>
      <option value="4">LSTM</option>
    </select>
<center>
    <button type="submit" style="background-color: #7ce780; color: rgb(20, 19, 19); padding: 10px 20px; border: none; border-radius: 3px; cursor: pointer; margin-top: 10px;">Submit</button>
  </center>
  </form>
```

```
 {% if path %}
 <div style="margin-top: 20px; text-align: center;">
   <p style="color: #cfbfbf;">Prediction: {{ prediction }}</p>
 </div>
 {% endif %}



 {% if message %}
 <script>
     alert("{{message}}")
 </script>
{% endif %}
</div>

{% endblock %}
```

# NEW.HTML

```
<!DOCTYPE html>
<html>

<head>
 <!-- Basic -->
 <meta charset="utf-8" />
 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
 <!-- Mobile Metas -->
 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
 <!-- Site Metas -->
 <meta name="keywords" content="" />
 <meta name="description" content="" />
 <meta name="author" content="" />

 <title>CREDIT CARD</title>

 <!-- slider stylesheet -->
```

```
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/OwlCarousel2/2.3.4/assets/owl.carousel.min.css" />

  <!-- bootstrap core css -->
  <link rel="stylesheet" type="text/css" href="/static/css/bootstrap.css" />

  <!-- fonts style -->
  <link href="https://fonts.googleapis.com/css?family=Poppins:400,600,700&display=swap" rel="stylesheet">
  <!-- Custom styles for this template -->
  <link href="/static/css/style.css" rel="stylesheet" />
  <!-- responsive style -->
  <link href="/static/css/responsive.css" rel="stylesheet" />
</head>

<body style="background-color: red;">
  <div class="hero_area">
    <!-- header section strats -->
    <header class="header_section">

      <div class="container-fluid">
        <nav class="navbar navbar-expand-lg custom_nav-container ">
          <span style="color: rgb(237, 244, 250); font-size: 25px;">
            <u style="color: rgb(255, 254, 252);">
            PREDICTING CREDIT CARD FRAUD DETECTION
           </u>
           </span>
          </a>
          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
          </button>
          <div class="collapse navbar-collapse" id="navbarSupportedContent">
            <div class="d-flex ml-auto flex-column flex-lg-row align-items-center">

              <ul class="navbar-nav  ">
                <li class="nav-item ">
                  <a class="nav-link" href="{{url_for('Home')}}">Home</a>
                </li>
        <li class="nav-item">
         <a class="nav-link" href="{{url_for('model')}}">model</a>
        </li>
        <li class="nav-item">
         <a class="nav-link" href="{{url_for('Prediction')}}">Prediction</a>
        </li>
        <li class="nav-item">
         <a class="nav-link" href="{{url_for('Home')}}">Logout</a>
        </li>
        </ul>
      </div>
     </div>
    </nav>
   </div>
  </header>

{% block content %}
  <!-- end header section -->
  <!-- slider section -->
  <section class=" slider_section position-relative">
   <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
    <div class="carousel-inner">


     <div class="carousel-item active">
      <div class="container">
       <div class="col-lg-10 col-md-11 mx-auto">
        <div class="detail-box">
         <div>
          <h5>
           PREDICTING
          </h3>
          <h4>
           CREDIT CARD FRAUD DETECTION
          </h2>
          <h2>
           USING MACHINE LEARNING
          </h2>
```

```
                  <p style="font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
font-size: 20px;">
                    Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both
institutions and consumers against financial losses. The increasing sophistication of fraudulent activities poses a
significant challenge, necessitating advanced detection methods to combat evolving threats. Credit card fraud
encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation.
These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics
employed by cybercriminals.
                  </p>
                </div>
              </div>
            </div>
          </div>
        </div>

        <!-- New slider item -->
        <div class="carousel-item">
          <div class="container">
            <div class="col-lg-10 col-md-11 mx-auto">
              <div class="detail-box">
                <div>
                  <h5>
                    FRAUD DETECTION
                  </h5>
                  <h4>
                    SECURING FINANCIAL TRANSACTIONS
                  </h4>
                  <h2>
                    WITH AI TECHNOLOGIES
                  </h2>
                  <p style="font-family: 'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida Grande', 'Lucida Sans', Arial, sans-serif;
font-size: 20px;">
                    Leveraging AI technologies for fraud detection not only enhances accuracy but also speeds up the
identification process. Implementing machine learning models trained on extensive datasets allows for real-time
analysis and prompt detection of fraudulent activities, minimizing financial losses and ensuring the integrity of financial
systems.
                  </p>
                </div>
            </div>
           </div>
          </div>
         </div>

        </div>
        <!-- Slide buttons -->
        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
          <span class="carousel-control-prev-icon" aria-hidden="true"></span>
          <span class="sr-only">Previous</span>
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
          <span class="carousel-control-next-icon" aria-hidden="true"></span>
          <span class="sr-only">Next</span>
        </a>
      </div>
    </section>
{% endblock %}

  <script src="/static/js/jquery-3.4.1.min.js"></script>
  <script src="/static/js/bootstrap.js"></script>

</body>
</html>
```

# PREDICTION.HTML

```
{% extends 'Home.html' %}

{% block navbar %}

<ul class="navbar-nav  ">
  <li class="nav-item ">
    <a class="nav-link" href="{{url_for('Home')}}">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="{{url_for('model')}}">model </a>
  </li>
  <li class="nav-item active">
  <a class="nav-link" href="{{url_for('Prediction')}}"> Prediction</a>
 </li>
 <li class="nav-item">
  <a class="nav-link" href="{{url_for('index')}}"> Logout</a>
 </li>
 </ul>
{% endblock %}
 {% block content %}
 <body style="font-family: Arial, sans-serif; background-image: url('/static/images/trainer-bg.jpg'); background-size: cover; background-position: center; height: 100vh; width: 100%; margin: 0; padding: 0;">
   <div style="width: 80%; height: 1050px; max-width: 400px; margin: 10px auto; padding: 20px; background-color: rgba(247, 241, 241, 0.212); border-radius: 8px; transition: all 0.3s ease;">

     <h1 style="text-align: center; margin-bottom: 20px; color: rgba(201, 201, 201, 0.692);">Prediction</h1>

     {% if result %}
      <h3 style="color: rgb(247, 247, 240); text-align: center;">Prediction: {{result}}</h3>
     {% endif %}

     <form action="{{url_for('Prediction')}}" method="post" style="display: flex; flex-direction: column;">
      <label for="category" style="margin-bottom: 1px; font-weight: bold; color: rgb(7, 6, 6);">category</label>
      <select id="category" name="category" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>
       <option >Choose Option</option>
       <option value="0">entertainment</option>
        <option value="1">food_dining</option>
        <option value="2">gas_transport</option>
        <option value="3">grocery_net</option>
        <option value="4">grocery_pos</option>
        <option value="5">health_fitness</option>
        <option value="6">home</option>
        <option value="7">kids_pets</option>
        <option value="8">misc_net</option>
        <option value="9">misc_pos</option>
        <option value="10">personal_care</option>
        <option value="11">shopping_net</option>
        <option value="12">shopping_pos</option>
        <option value="13">travel</option>
```

```
    </select>

     <!-- Add more options as needed -->
    </select>
    <label for="amt" style="margin-bottom: 1px; font-weight: bold; color: rgb(10, 10, 10);">Amount:</label>
    <input type="float" id="amt" name="amt" step="0.0000000000001" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>

    <label for="gender" style="margin-bottom: 1px; font-weight: bold;color: rgb(15, 15, 14);">Gender :</label>
    <select id="gender" name="gender" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>
     <option >Choose Option</option>
      <option value="0">Female</option>
      <option value="1">Male</option>
      <!-- Add more options as needed -->
    </select>

    <label for="top_5_city" style="margin-bottom: 1px; font-weight: bold;color: rgb(20, 19, 16);">top_5_city:</label>
    <select id="top_5_city" name="top_#5_city" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>
     <option >Choose Option</option>
     <option value="0">Birmingham </option>
      <option value="1">Conway</option>
      <option value="2">Meridian </option>
      <option value="4">Phoenix</option>
      <option value="5">San Antonio </option>
      <option value="6">Utica</option>
      <option value="3">Other</option>
      <!-- Add more options as needed -->
    </select>

    <label for="top_5_job" style="margin-bottom: 1px; font-weight: bold;color: rgb(31, 30, 29);">top_5_jobs:</label>
    <select id="top_5_job" name="top_5_job" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>
     <option >Choose Option</option>
     <option value="0">Exhibition designer </option>
     <option value="1">Film/video editor</option>
```

```
          <option value="2">Materials engineer </option>
          <option value="3">Naval architect</option>
          <option value="5">Surveyor, land/geomatics </option>
          <option value="6">Systems developer</option>
          <option value="4">Other</option>
            <!-- Add more options as needed -->
        </select>
        <label for="top_5_in_merchant" style="margin-bottom: 1px; font-weight: bold;color: rgb(20, 20,
20);">top_5_in_merchants:</label>
        <select id="top_5_in_merchant" name="top_5_in_merchant" style="margin-bottom: 5px; padding: 10px; font-
size: 16px; border: 1px solid #abf0bc; border-radius: 4px;" required>
          <option >Choose Option</option>
          <option value="1">fraud_Boyer PLC</option>
          <option value="2">fraud_Cormier LLC </option>
          <option value="3">fraud_Dickinson Ltd</option>
          <option value="4">fraud_Kilback LLC </option>
          <option value="5">Sfraud_Kuhn LLC</option>
          <option value="6">fraud_Schumm PLC</option>
          <option value="0">Other </option>
            <!-- Add more options as needed -->
        </select>
        <label for="age" style="margin-bottom: 1px; font-weight: bold; color: rgb(10, 10, 10);">Age:</label>
        <input type="number" id="age" name="age" step="0.01" style="margin-bottom: 10px; padding: 10px; font-size:
16px; border: 1px solid #abf0bc; border-radius: 4px;" required>

        <!-- <label for="Age" style="margin-bottom: 1px; font-weight: bold;color: rgb(29, 28, 27);">Age:</label>
        <select id="Age" name="Age" style="margin-bottom: 5px; padding: 10px; font-size: 16px; border: 1px solid
#abf0bc; border-radius: 4px;" required>
            <option value="8400000">8,400,000</option>
            <option value="4000000">4,000,000</option> -->
            <!-- Add more options as needed -->
        </select>



            <!-- Add more options as needed -->
        </select>
        <button type="submit" style="padding: 10px 20px; font-size: 16px; color: #fff; background-color: #00000079; border: none; border-radius: 4px; cursor: pointer;">Submit</button>
      </form>

    </div>
 </body>
 </html>

  <script src="js/jquery-3.4.1.min.js"></script>
  <script src="js/bootstrap.js"></script>
{% endblock %}
```

# REGISTER.HTML

```
{% extends 'index.html' %}

{% block navbar %}
  <li><a href="{{url_for('index')}}">Home</a></li>
  <li><a href="{{url_for('about')}}">About</a></li>
  <li class="active"><a href="{{url_for('register')}}">Register</a></li>
  <li><a href="{{url_for('login')}}">Login</a></li>
{% endblock %}

{% block content %}
  <center>
    <section class="section coming-soon" data-section="section3">
      <div class="col-5">
        <form id="contact" action="{{url_for('register')}}" method="post">
          <center>
            <h1 style="color: white;">Registration</h1><br>
          </center>
          <div class="row">

            <div class="col-md-12">
              <fieldset>
                <input name="name" type="text" class="form-control" id="name" placeholder="User Name" required="">
              </fieldset>
            </div>

            <div class="col-md-12">
              <fieldset>
                <input name="email" type="email" class="form-control" id="email" placeholder="Email ID"
required="">
              </fieldset>
            </div>

            <div class="col-md-12">
              <fieldset>
                <input name="password" type="password" class="form-control" id="password"
placeholder="Password" required="">
              </fieldset>
            </div>

            <div class="col-md-12">
              <fieldset>
                <input name="c_password" type="password" class="form-control" id="password"
placeholder="Conform Password" required="">
              </fieldset>
            </div>

            <div class="col-md-12">
              <fieldset>
                <button type="submit" id="form-submit" class="button" style="color: black;">Submit</button>
              </fieldset>
            </div>
          </div>
        </form>
      </div>
    </section>
  </center>
{% endblock %}
```

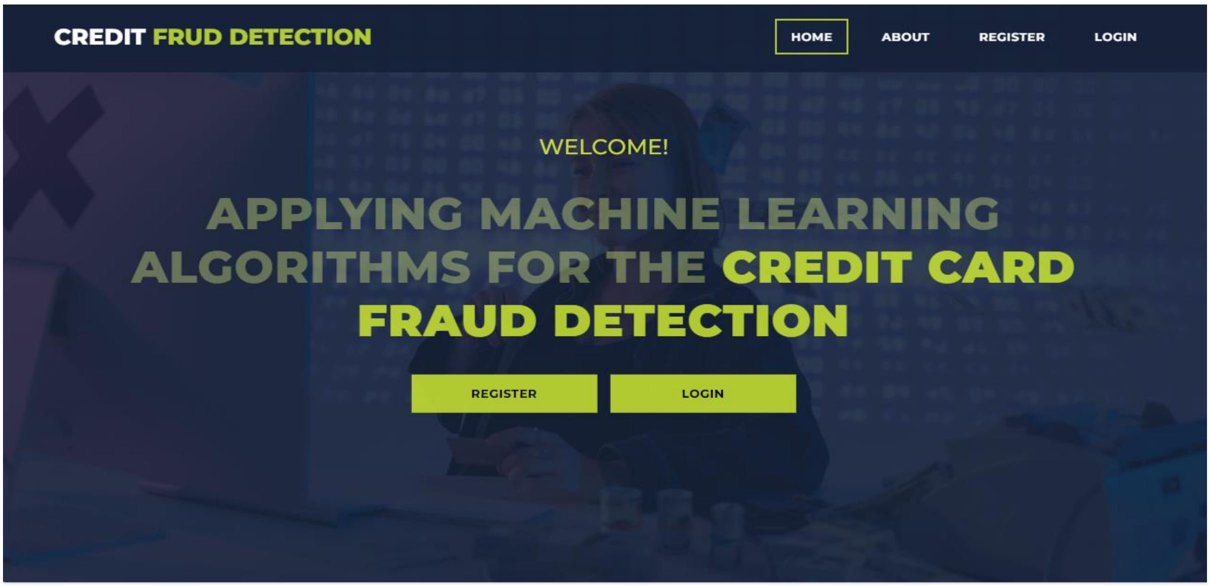
# BACK-END CODE

# APP.PY

```
from flask import Flask,url_for,render_template,request
import joblib
import mysql.connector,os ,re , joblib

app = Flask(__name__)



mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    port="3307",
    database='db'
)
mycursor = mydb.cursor()

def executionquery(query,values):
    mycursor.execute(query,values)
    mydb.commit()
    return

def retrivequery1(query,values):
    mycursor.execute(query,values)
    data = mycursor.fetchall()
    return data

def retrivequery2(query):
    mycursor.execute(query)
    data = mycursor.fetchall()
    return data
@app.route("/")
def index():
    return render_template('index.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/register', methods=["GET", "POST"])
def register():
    if request.method == "POST":
        email = request.form['email']
        name = request.form['name']
        password = request.form['password']
        c_password = request.form['c_password']
        if password == c_password:
            query = "SELECT UPPER(email) FROM users"
            email_data = retrivequery2(query)
            email_data_list = []
            for i in email_data:
                email_data_list.append(i[0])
            if email.upper() not in email_data_list:
                query = "INSERT INTO users (name,email, password) VALUES ( %s, %s, %s)"
                values = (name,email, password)
                executionquery(query, values)
                return render_template('login.html', message="Successfully Registered!")
            return render_template('register.html', message="This email ID is already exists!")
        return render_template('register.html', message="Conform password is not match!")
    return render_template('register.html')

@app.route('/login',methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password']

        query = "SELECT UPPER(email) FROM users"
        email_data = retrivequery2(query)
        email_data_list = []
        for i in email_data:
            email_data_list.append(i[0])

        if email.upper() in email_data_list:
            query = "SELECT UPPER(password) FROM users WHERE email = %s"
            values = (email,)
            password__data = retrivequery1(query, values)
            if password.upper() == password__data[0][0]:
                global user_email
                user_email = email

                return render_template('home.html')
            return render_template('login.html', message= "Invalid Password!!")
        return render_template('login.html', message= "This email ID does not exist!")
    return render_template('login.html')




@app.route('/Home')
def Home():
    return render_template('Home.html')

@app.route('/Prediction', methods = ['GET',"POST"])
def Prediction():
    if request.method == 'POST':
        category = int(request.form['category'])
        amt = (request.form['amt'])
        top_5_city = int(request.form['top_5_city'])
        gender = int(request.form['gender'])
        top_5_job = int(request.form['top_5_job'])
        top_5_in_merchant = float(request.form['top_5_in_merchant'])
        age = float(request.form['age'])

        lee = [[category, amt, top_5_city,gender, top_5_job, top_5_in_merchant,age]]
        print(lee)
        print(lee)
        model = joblib.load("randomforest.joblib")
        predictions = model.predict(lee)
        print(predictions)
        if predictions == 0:
            result = 'No Fraud'
            return render_template('prediction.html', result = result)
        elif predictions == 1:
            result = 'Fraud'
            return render_template('prediction.html', result = result)
    return render_template('prediction.html')

@app.route('/model',methods = ['GET',"POST"])
def model():
    msg = None
    accuracy = None
    if request.method == 'POST':
      algorithams = request.form["algorithm"]
      if algorithams == "1":
         accuracy = 99
         msg = 'Accuracy  for Stacking Classifier is ' + str(accuracy) + str('%')
      elif algorithams == "2":
         accuracy = 99
         msg = 'Accuracy  for RandomForestClassifier is ' + str(accuracy) + str('%')
      elif algorithams == "3":
         accuracy = 88
         msg = 'Accuracy  for Convolutional Neural Network(CNN) is ' + str(accuracy) + str('%')
      elif algorithams == "4":
         accuracy = 92
         msg = 'Accuracy  for Long Short-Term Memory(LSTM) is ' + str(accuracy) + str('%')
      return render_template('model.html',msg=msg,accuracy = accuracy)
    return render_template('model.html')

if __name__ == "__main__":
    app.run(debug=True)
```

4.5 OUTPUT Screens:

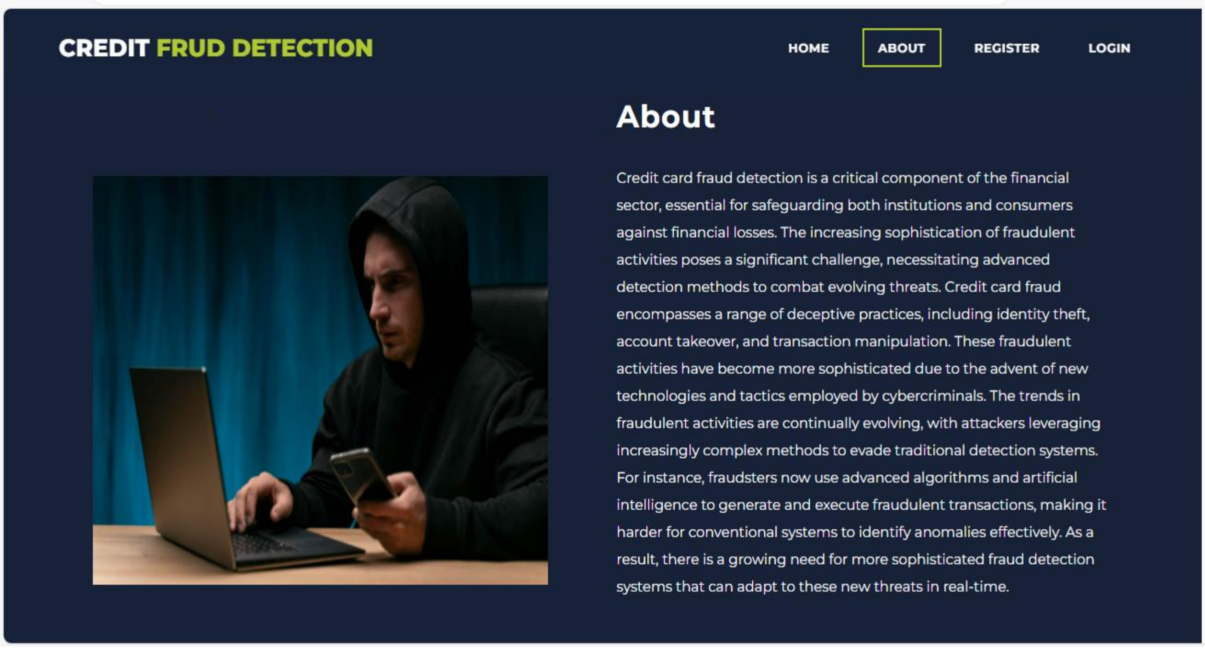
HOME PAGE : Home page of the project



WELCOME!

APPLYING MACHINE LEARNING ALGORITHMS FOR THE CREDIT CARD FRAUD DETECTION

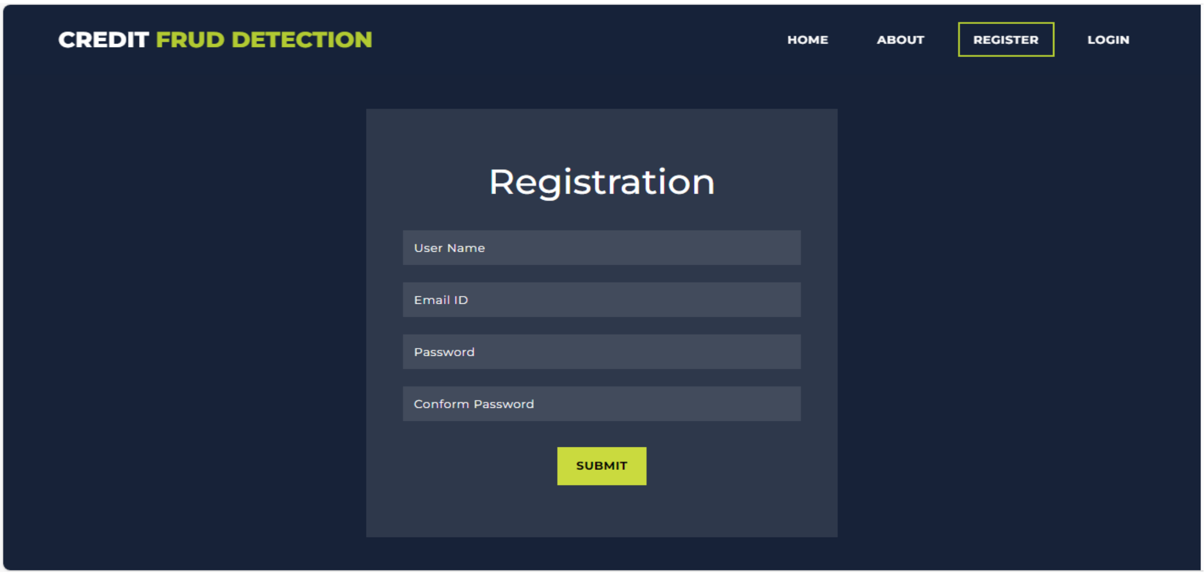
About Page:



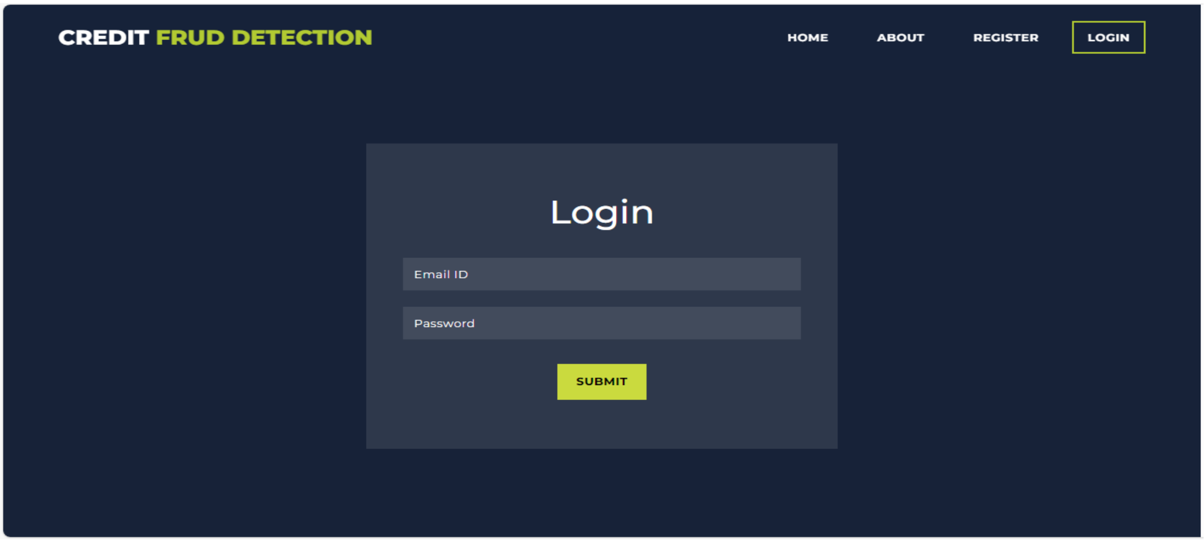
About

Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both institutions and consumers against financial losses. The increasing sophistication of fraudulent activities poses a significant challenge, necessitating advanced detection methods to combat evolving threats. Credit card fraud encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation. These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics employed by cybercriminals. The trends in fraudulent activities are continually evolving, with attackers leveraging increasingly complex methods to evade traditional detection systems. For instance, fraudsters now use advanced algorithms and artificial intelligence to generate and execute fraudulent transactions, making it harder for conventional systems to identify anomalies effectively. As a result, there is a growing need for more sophisticated fraud detection systems that can adapt to these new threats in real-time.

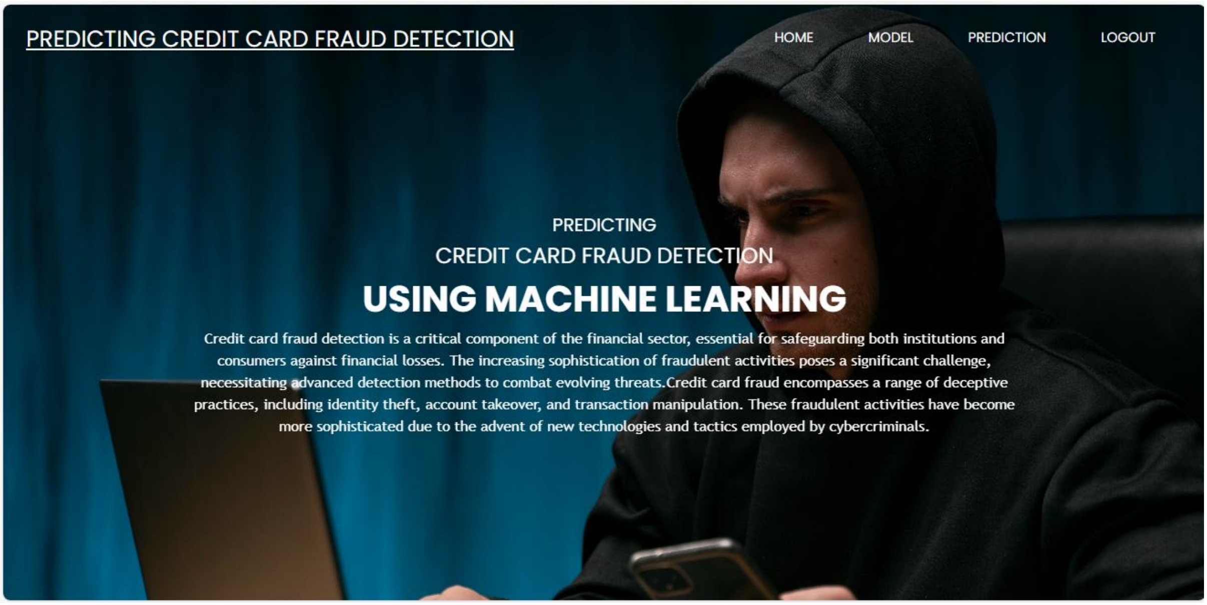
Registration Page:



Login Page:



Model Home Page

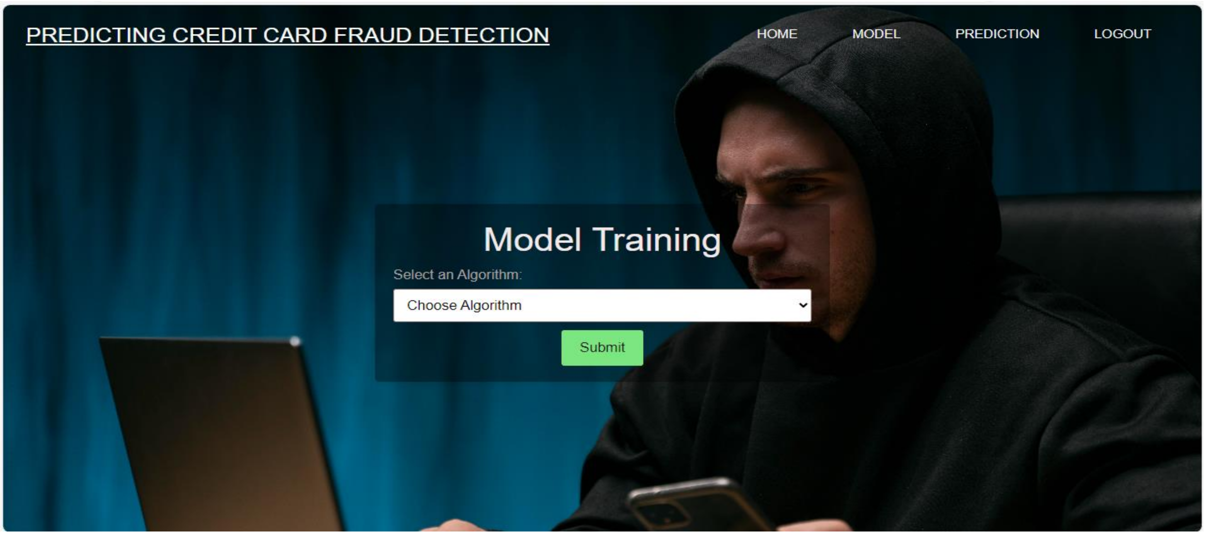


PREDICTING CREDIT CARD FRAUD DETECTION

PREDICTING
CREDIT CARD FRAUD DETECTION
USING MACHINE LEARNING

Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both institutions and consumers against financial losses. The increasing sophistication of fraudulent activities poses a significant challenge, necessitating advanced detection methods to combat evolving threats.Credit card fraud encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation. These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics employed by cybercriminals.

Model Selection page:

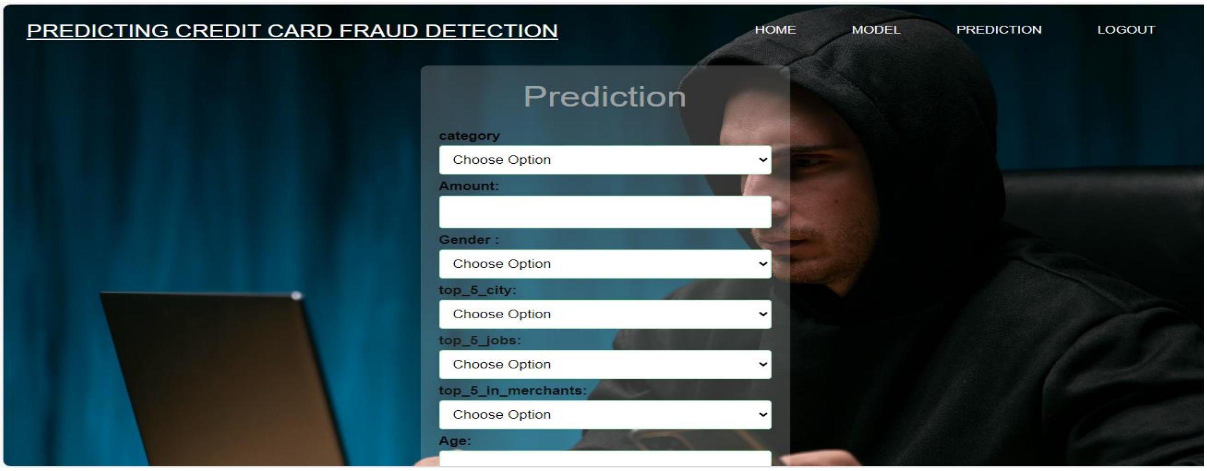


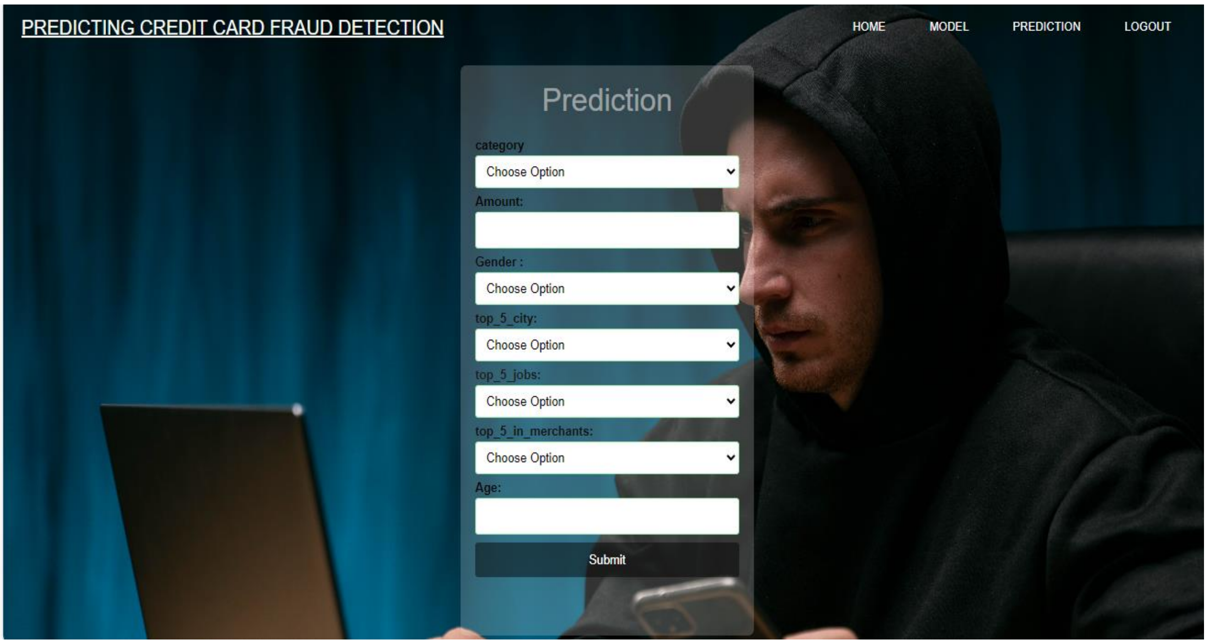
PREDICTING CREDIT CARD FRAUD DETECTION

Model Training

Select an Algorithm: Choose Algorithm

Prediction page:





CHAPTER 5 : SYSTEM TESTING

5.SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1 TYPES OF TESTING

5.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.1.2 Integration Testing

- **Purpose:** Verifies that integrated components function together as a single system.
- **Focus:** Ensures smooth interaction between components after unit testing.
- **Goal:** Identifies issues arising from the combination of different software parts.
- **Outcome:** Confirms consistency and proper integration of all system elements.

5.1.3 Functional Testing

Functional testing checks that the software works as intended, based on business requirements and documentation. It focuses on:

- **Valid Inputs:** Ensuring the system accepts only correct data.
- **Invalid Inputs:** Rejecting incorrect data without errors.
- **Core Functions:** Verifying essential functions perform as expected.
- **Output Accuracy:** Ensuring the system generates the right outputs.
- **System Interactions:** Confirming smooth integration with other systems.

Tests are designed to cover key business processes, data fields, and workflows, ensuring everything functions perfectly before final validation.

5.1.4 System testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.1.5 White Box testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

5.1.6 Black Box testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot ―seeǁ into it. The test provides inputs and responds to outputs without considering how the software works.

5.1.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered

5.2 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail**.**

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.4 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 6 : CONCLUSION

6. Conclusion

Summary of Key Findings

The intention of this study was to compare four ml algorithms for fraud detection. Findings indicate that each of the four algorithms, namely CNN, LSTM, Decision Trees, and Random Forests, demonstrated promising performance in identifying fraudulent transactions.

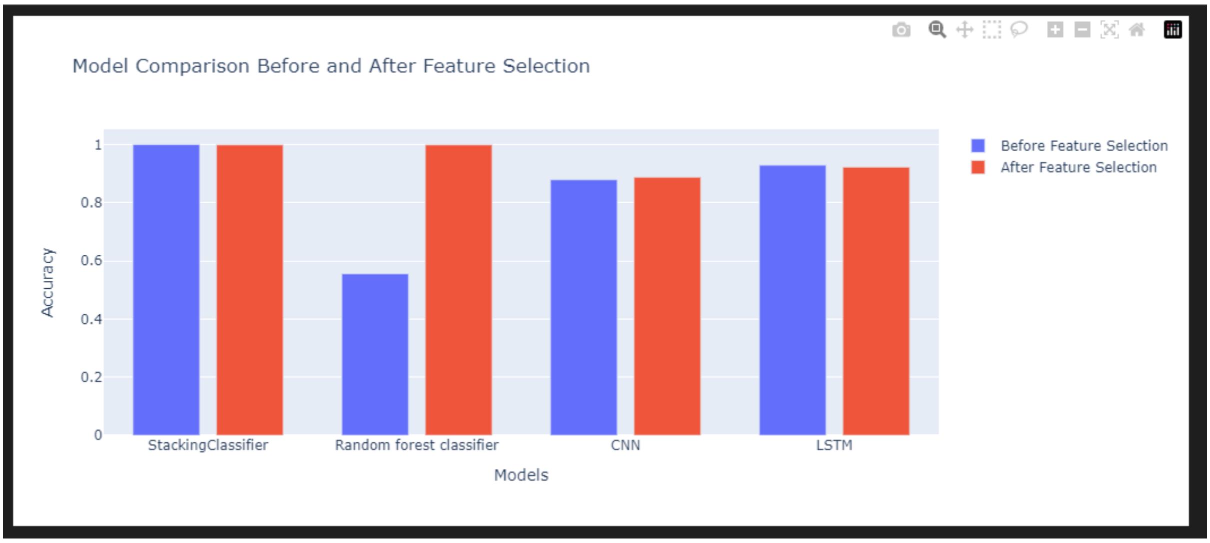


Figure 6 Accuracy comparison

Key Results:

CNNs achieved an accuracy of 88.6% in detecting fraud, outperforming the other algorithms. This suggests that CNNs are effective in capturing intricate patterns in transaction data.

LSTM showed an accuracy of 92.2%, indicating their ability to model sequential dependencies and temporal dynamics in transaction data.

Decision Trees and Random Forests achieved accuracies of 98.5% and 99.1%, respectively, demonstrating their robust classification capabilities.

Implications:

The findings imply that ml algorithms possess the capacity to significantly increase the precision and effectiveness of fraud detection systems. The comparative analysis of the algorithms highlights the importance of finding the most suitable algorithm for a particular problem, as different algorithms excel in different aspects of fraud detection.

The study's findings have significant implications for the development of more secure credit card transaction systems, which can mitigate financial losses and enhance consumer trust.

Conclusion

This study highlights the significant potential of machine learning algorithms in improving fraud detection systems. Through a comparative analysis of CNN, LSTM, Decision Trees, and Random Forests, it was demonstrated that these algorithms can effectively identify fraudulent transactions with varying levels of accuracy. The findings emphasize the importance of selecting the right algorithm based on the specific requirements of the task, as each algorithm excels in different aspects of fraud detection.

The study contributes to the ongoing development of more secure credit card transaction systems, offering insights into the integration of machine learning models that can help mitigate financial losses and enhance consumer trust. However, there are still opportunities for further advancements, such as the integration of more sophisticated algorithms, real-time detection systems, continuous model updates, and enhanced feature engineering. These future enhancements will make fraud detection systems more adaptive, precise, and effective in combating increasingly sophisticated fraudulent activities.

By implementing the recommendations and exploring new research directions, financial institutions can improve their fraud detection capabilities, reduce losses, and foster greater consumer confidence in their securitymeasures.

6. Future Enhancements for Fraud Detection Systems

1. **Integration of Advanced Algorithms**

   Future research could explore incorporating more sophisticated algorithms, such as boosting techniques and reinforcement learning, to continuously adapt fraud detection models based on real-time feedback. CNNs could also be used to generate synthetic fraud patterns, improving the model's ability to detect emerging fraud tactics.

2. **Feature Engineering and Selection**

   Enhancing feature engineering with additional data sources like geolocation and social media activity could improve model performance. Utilizing advanced feature selection techniques will help in identifying the most relevant features, reducing noise, and improving prediction accuracy.

3. **Real-Time Detection and Response Systems**

   Developing real-time fraud detection systems that can instantly flag suspicious transactions is essential. This requires fast algorithms and a robust infrastructure for quick data processing. Exploring cloud and edge computing for real-time processing could enhance fraud prevention.

4. **Continuous Model Training and Updates**

   To keep up with evolving fraud patterns, continuous retraining of models with fresh transaction data is critical. This will ensure the models stay effective in detecting new fraud tactics. Incorporating a feedback loop for flagged transactions can improve accuracy over time.

5. **Exploring Additional Algorithms**

   In future research, it would be beneficial to compare more advanced machine learning methods, such as ensemble techniques, XGBoost, and gradient boosting machines (GBM), to explore whether they offer better performance than the four algorithms studied.

6. **Enhanced Evaluation Metrics**

   Incorporating additional evaluation metrics like F1-score, AUC-ROC, and Matthews Correlation Coefficient (MCC) could offer deeper insights, especially for imbalanced datasets typical in fraud detection, where fraudulent transactions are much less frequent than legitimate ones.

REFERENCES

1. Adi Saputra1, Suharjito2L: Fraud Detection using Machine Learning in e-Commerce, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 9, 2019.

2. Dart Consulting, Growth of Internet Users in India and Impact on Country's Economy: https://www.dartconsulting.co.in/marketnews/growth-of-internet-users-in-india-and-impact-on-countryseconomy/

3. Ganga Rama Koteswara Rao and R.Satya Prasad, " -Shielding The Networks Depending On Linux Servers Against Arp Spoofing, International Journal of Engineering and Technology(UAE),Vol. 7, PP.75-79, May 2018, ISSN No: 2227-524X, DOI - 10.14419/ijet.v7i2.32.13531.

4. Heta Naik , Prashasti Kanikar: Detctionsbased on Machine Learning Algorithms,International Journal of Computer Applications (0975 – 8887) Volume 182 – No. 44, March 2019.

5. Navanshu Khare ,Saad Yunus Sait: DetctionsUsing Machine Learning Models and Collating Machine Learning Models, International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 825-838 ISSN: 1314-3395.

6. Randula Koralage, , Faculty of Information Technology, University of Moratuwa,Data Mining Techniques for Credit Card Fraud Detection.

7. Roy, Abhimanyu, et al:Deep learning detecting fraud in credit card transactions, 2018 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018.

8.Sahayasakila.V, D. Kavya Monisha, Aishwarya, Sikhakolli VenkatavisalakshiseshsaiYasaswi: DetctionsSystem using Smote Technique and Whale Optimization Algorithm,International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

9. Statista.com. retail e-commerce revenue forecast from 2017 to 2023 (in billion U.S. dollars). Retrieved April 2020, from India : https://www.statista.com/statistics/280925/e-commercerevenueforecast-in-india/

10. Yashvi Jain, NamrataTiwari, Shripri yaDubey, Sarika Jain:A Comparative Analysis of Various DetctionsTechniques, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2.

FORCASTING CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

[1] P. Veeresh, [2] M. Thaher Basha, [3] Vajrala Varshith, [4] Beldar Taher Basha, [5] Kadapala Giribabu

[1] Associate professor, [2, 3, 4, 5,] BTECH

[1, 2, 3, 4, 5,] Dept of CSE, St. Johns College of Engineering and Technology, Yerrakota, Yemmiganur, Kurnool, AP, Affiliated by JNTUA, INDIA

[1] taherb753@gmail.com, [2] vajralavarshith@gmail.com, [3] taherbasha506@gmail.com,

[4] kadapalagiribabu2003@gmail.com,

ABSTRACT

Identification of bank cards fraud remains a serious worry for financial companies because of how sophisticated criminal behavior is becoming.. This study addresses this challenge by leveraging models increase the efficiency of detection systems. Utilizing a dataset from Kaggle, we implement and compare five distinct algorithms: LSTM networks, neural networks based on CNN, Decision Trees, Random Forests, and a Stacking Classifier. CNNs are employed to capture intricate patterns in transaction data, while LSTM address sequential dependencies and temporal dynamics. Decision Trees and Random Forests provide robust classification through hierarchical decision-making and ensemble learning. Additionally, a Stacking Classifier integrates the strengths of these algorithms to potentially improve overall performance. The comparative analysis of these methods tries to determine the best method for real-time identification of fraud. The results are expected to contribute significantly to the development of more secure credit card transaction systems, thereby mitigating financial losses and enhancing consumer trust.

1. INTRODUCTION

Overview

Credit card fraud detection is a vital aspect of the financial industry, designed to protect both institutions and consumers from financial losses. As fraudulent activities become more sophisticated, traditional detection methods are struggling to keep pace with the evolving tactics used by cybercriminals. Fraudulent practices like identity theft, account takeover, and transaction manipulation are now being carried out using advanced technologies and algorithms, making detection increasingly difficult.

Fraudsters employ complex techniques, including artificial intelligence, to bypass conventional detection systems, highlighting the need for more advanced, adaptable fraud detection mechanisms. These sophisticated threats demand real-time, dynamic detection systems to identify and combat fraud effectively.

The economic consequences of credit card fraud are severe, leading to substantial losses for financial institutions and damage to their reputations. Consumers are also impacted by financial losses and emotional distress. Consequently, there is a pressing need for more effective and advanced fraud detection systems that can evolve alongside these new and complex threats.

Problem Statement

Fraud detection remains A critical concern for financial organazations, as traditional methods struggle to keep pace with the rapid evolution of fraudulent tactics. Conventional methods for detecting fraud frequently depend on rule-based techniques, which involve predefined criteria and patterns for identifying suspicious activities. While these systems can be effective against known types of fraud, they are inherently limited by their static nature. They often fail to detect new, sophisticated fraud schemes that do not match the established rules or patterns. This limitation leaves significant gaps in fraud prevention, allowing innovative fraudulent activities to go undetected. Another significant challenge is the dynamic Fraudsters are constantly evolving their techniques and strategies to carry out fraudulent activities. They consistently modify and refine their methods to stay ahead of detection systems.

Objective of the project

Main goal of this investigation is to leverage cutting-edge ML techniques to enhance the accuracy and efficiency of detection systems. Fraud remains a major challenge. For financial institutions because of their increasing sophistication of fraudulent activities. Traditional techniques for detecting cheating often fall.

CNN: are employed to capture intricate patterns and anomalies within transaction data. CNNs excel in identifying complex features through their hierarchical layers, making them suitable for detecting subtle signs of fraud.

LSTM : are utilized to address sequential dependencies and temporal dynamics inherent in transaction data. LSTM are particularly effective in analyzing time-series data, which is crucial for detecting fraud patterns that evolve over time.

Decision Trees: provide a transparent, hierarchical approach to classification, making decisions based on feature values. They offer interpretability and are useful for understanding decision rules.

Random Forests: build upon Decision Trees by employing an ensemble learning approach. This method aggregates using choice trees to enhance classification accuracy. Accuracy and reduce overfitting, enhancing the overall robustness of the fraud detection system

Limitations of the Project

This study employs a Kaggle dataset to Create and assess ML models for the detection of credit card fraud using the dataset provided. a critical component of this research, comprises transaction records with labeled instances of fraudulent and legitimate activities. The dataset encompasses multiple attributes, including the transaction amount and timestamp. and user-specific details, which are essential for training and testing the algorithms.

2 LITERATURE SURVEY

1. R. Sailusha, V. Gnaneswar, R. Ramesh, and G. R. Rao (2022) Title: Credit Card Fraud Detection Using Machine Learning

This study compares two popular machine learning algorithms, Random Forest and Adaboost, for credit card fraud detection. The paper evaluates both algorithms based on several metrics, including accuracy, precision, recall, F1-score, and the ROC curve. Through this comparison, the authors aim to determine which method is the most effective for detecting fraud in credit card transactions.

2. A. Thennakoon, C. Bhagvani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarac (2019)

Title: Real-time Credit Card Fraud Detection Using Machine Learning

This paper explores the use of machine learning models for real-time credit card fraud detection. The authors evaluate optimal algorithms for detecting various types of fraud and implement a system for real-time fraud detection through predictive analytics. An API is also used to facilitate the real-time application of the system, providing efficient fraud detection solutions.

3 1. J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare (2017) Title: Credit card fraud detection using machine learning techniques: A comparative analysis

In this paper, the authors highlight the critical role of data mining in detecting credit card fraud. The study emphasizes the challenges posed by adaptive detection mechanisms and handling imbalanced datasets, particularly when working with real-time transaction data. Through a comparative analysis of various machine learning techniques, the paper offers insights into how data mining can enhance the accuracy and efficiency of fraud detection systems

.

4. F. K. Alarfaj, L. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Aluned (2022) Title: Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms

This paper introduces a hybrid system combining an autoencoder with a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) for credit card fraud detection. The proposed model aims to address the challenges of detecting financial fraud with high accuracy, particularly in the context of imbalanced datasets. The authors use oversampling techniques to balance the dataset, which significantly improves both recall and precision in fraud detection, making it a highly effective solution in real-world applications.

5 D. Varmedia, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla (2019) Title: Credit Card Fraud Detection Using Machine Learning Methods

In this research, the authors employ various machine learning techniques to detect credit card fraud. They use SMOTE (Synthetic Minority Over-sampling Technique) for balancing the dataset, which helps in dealing with the common problem of imbalanced classes in fraud detection tasks. The paper also applies feature selection to enhance the

performance of machine learning models, including Logistic Regression, Random Forest, Naive Bayes, and Multilayer Perceptron (MLP). Their approach demonstrates high accuracy in detecting fraud despite the challenges of working with imbalanced datasets.

3 - SYSTEM ANALYSIS

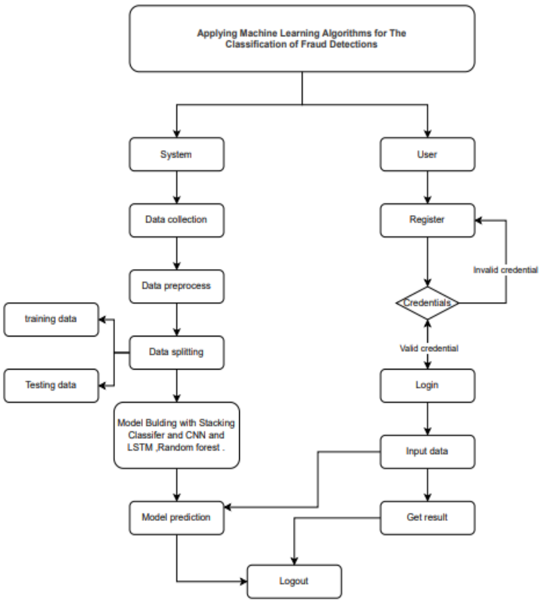
Overview of Existing Model:

Current credit card fraud detection systems primarily rely on rule-based methods, statistical models, and traditional machine learning algorithms such as Logistic Regression and Support Vector Machines (SVM). These systems often face limitations in detecting complex and evolving fraud patterns due to their reliance on static rules and the inability to capture temporal dependencies in transaction data. As fraudsters develop more sophisticated techniques, these systems struggle to maintain high accuracy and timely detection, leading to false positives and false negatives, which can result in either customer dissatisfaction or financial losses for institutions

Overview of Proposed Model:

The proposed method for enhancing credit card fraud detection involves the implementation and comparison of five machine learning algorithms: Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), Decision Trees, Random Forests, and a Stacking Classifier. Each algorithm is chosen for its unique strengths in handling complex transactional data. CNNs are utilized to identify intricate patterns within the data, while LSTMs are employed to capture sequential dependencies and temporal relationships. Decision Trees offer straightforward, interpretable decision-making, and Random Forests enhance this by aggregating multiple trees to improve classification accuracy. The Stacking Classifier integrates predictions from the CNN, LSTM, Decision Trees, and Random Forests to create a more robust and comprehensive model. This method aims to optimize detection accuracy and efficiency, thereby reducing false positives and improving real-time detection capabilities in credit card fraud prevention systems.

Workflow of Proposed Model



4. **METHODOLOGY**

Dataset Description

1Unnamed: 0 ==> This feature represents Likely an index or row identifier from the dataset. This column might not hold meaningful data and can often be dropped.

2trans_date_trans_time ==> This Componet indicates The timestamp of the transaction, typically in a date-time notation (e.g., HH:MM:SS or YYYY-MM-DD).

3cc_num ==> This feature indicates The credit card number used for the transaction. This is sensitive information and should be handled with care 4.merchant ==> This aspect indicates The name of the merchant or store where the transaction occurred.

5.category ==> This attribute records the kind or category of products or services that were bought (e.g., groceries, electronics).

6.amt ==> it’s represents The amount of money spent in the transaction.

7. first ==> This section indicates The first name of the cardholder.

8. last ==> This feature captures The last name of the cardholder.

9. gender ==> it shows The gender of the cardholder (e.g., Male, Female).

10. street ==> The place of residence connected to the hyping information of the cardholder.

11.city ==> This aspect captures The merchant billing address city.

12.state ==> This feature indicates The status of the advertising address.

13.zip ==> This col represents The postal ZIP code associated with the cardholder's address.

14.lat ==> This component indicates The azimuth of the merchant's latitude coordinate.

15.long ==> This feature represents The geographic location reflected in the merchant's longitude coordinate.
16.city_pop ==> This variable represents The number of people residing in the city where the transaction took place.
17.job ==> This Columns indicates This feature representsThe occupation of the cardholder.
18.dob ==> it tells Record the cardholder's date of birth.
19. trans_num ==> This feature donates The unique transaction number or ID.
20.unix_time ==> it represents The timestamp of the transaction in Unix epoch time format.
21.merch_lat ==> This columns shows the latitude organize of where the merchant is located
22.merch_long ==> Column indicates the longitude coordinate of the holder location.
23.is_fraud ==> its represents A binary indicator (usually 0 or 1) showing whether the transaction was fraudulent (1) or not (0).

Data Collection

In this study, we utilize a dataset sourced from Kaggle, which is widely recognized for its extensive collection of data suitable for machine intelligence applications. The dataset is designed specifically for fraud detection and includes a diverse range of features essential for identifying fraudulent transactions. It comprises transaction details such as transaction amount, timestamp, and anonymized variables derived from credit card transactions, which capture various aspects of user behavior and transaction patterns.

Data Preprocessing

Effective data training is crucial for building robust subset of AI models, especially in the context of identificaton fraudwhere data quality can significantly impact model performance. The preprocessing stage involves several key steps:

Handling Missing Values:

Null values in the dataset are Located to ensure that the machine learning algorithms operate on complete data. Common methods include inference, in which statistical measurements are used to replace missing valuesSuch as the standard deviation, the median, or mode of operation, or more complex approaches. like k-Nearest Neighbors imputation. Alternatively, rows or columns with excessive missing values may be removed if they do not provide critical information.

Normalization:

To guarantee that different features contribute evenly to the model's performance., normalization is applied. This involves scaling feature values to a standard range, typically between 0 and 1 or -1 and 1. Methods such as Z-score Uniformity and Min-Max Scaling are commonly used. Normalization helps in stabilizing the training process and improving the convergence rate of algorithms like The networks made up of Long Short-Term Memory LSTM and Convolutional neural networks, more commonly are sensitive to the size of input features.

Feature Selection:

Selecting relevant features is essential to enhance model efficiency and accuracy. Feature selection methods, such as statistical tests, correlation analysis, and techniques like Recurring Structures Feature Elimination, or RFE or Principal Component Analysis (PCA), are used to identify and retain the most informative features while discarding redundant or irrelevant ones. This step helps in reducing dimensionality and mitigating the risk of overfitting.

Data Splitting:

The dataset is separated into training, validation, and test sets to evaluate model performance effectively. Typically, the data is split into Ten to fifteen percent for testing, seventy to eighty percent for training. The test set is used to gauge how well the final model can generalize, the validation set is used to adjust hyper parameters and monitor performance throughout training, and the training set is used to train the models.

5 Algorithm Implementation and Result

The implementation of algorithms for credit Detecting card theft requires a methodical approach to implementing CNN, LSTM, Decision Trees, and Random Forests. Each algorithm is tailored to address specific aspects of fraud detection and enhance the overall system's accuracy and efficiency.

CNN:

The CNN architecture is structured to achieve and detect complex patterns in transaction data. The implementation begins with defining the network's layers, including feature extraction using convolutional layers that pooling layers for dimensionality reduction, and fully connected layers for classification. The convolutional layers use filters to scan through the transaction data, identifying intricate patterns that might indicate fraudulent activity. Pooling layers help in reducing the computational load by summarizing the features extracted. In order to minimize the loss function, the network is trained using gradient descent and backpropagation. and optimize performance.

LSTM:

LSTM networks are configured to handle sequential data, capturing temporal dependencies crucial for detecting fraudulent patterns over time. The implementation involves setting up LSTM cells that maintain and update internal states through forget, input, and output gates. This structure allows the LSTM to remember long-term dependencies and recognize patterns in transaction sequences. Training involves using sequences of transaction data, with the LSTM adjusting its parameters to learn the temporal relationships between events.

Stacking Classifier

A Stacking Classifier is an ensemble learning method that blends several machine learning models rise predictive performance. Unlike traditional ensemble methods like bagging or boosting, which use a single type of model, stacking integrates different types of models to leverage their unique strengths.

Decision Trees and Random Forests:

To create decision forests, split data at each node to classify transactions, and build a hierarchical model that makes judgments based on feature values. By training an ensemble of decision trees using a randomized a portion of the characteristics and data, Random Forests expand on this concept. The forest aggregates The different trees' forecasts show increased robustness. and accuracy. Both algorithms are trained and evaluated using metrics such as accuracy, precision, and recall, to ensure effective fraud detection.

LIME Implementation

LIME (Local Interpretable Model-agnostic Explanations) is a technique that explains predictions of machine learning models. It works by approximating a complex model with a simpler, interpretable model locally around the prediction.

Model Training and Tuning

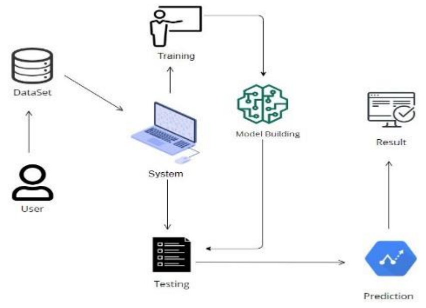
In this study, the model training process is designed to optimize the performance of five ML algorithms specifically for fraud detection. Each model undergoes extensive training using a preprocessed dataset from Kaggle, ensuring that the data is properly balanced to resolve the class disparity that exists in fraud detection activities.

For the CNN model, a grid search is conducted to fine-tune hyper parameters such as the number of convolutional layers, filter sizes, and learning rates. The LSTM model, given its ability to capture temporal dependencies, is optimized by adjusting the number of LSTM units, dropout rates, and sequence lengths. Decision Trees are pruned and tuned for depth and split criteria to avoid overfitting. Random Forests, being an ensemble of decision trees, are tuned by varying To find a Establish a balance between variation and bias, then calculate the maximum depth with all tree.

The Stacking Classifier combines the forecasts made by each separate model to generate a more reliable forecast. The base models for stacking are trained with optimal hyper parameters obtained from prior tuning processes. A meta-learner, typically a logistic regression or another tree-based model, is then trained on the outputs of these base models.

Cross-validation techniques are employed throughout the tuning process to validate the models' generalization performance, ensuring that the final models are not only accurate but also capable of handling real-world credit card transaction data efficiently.

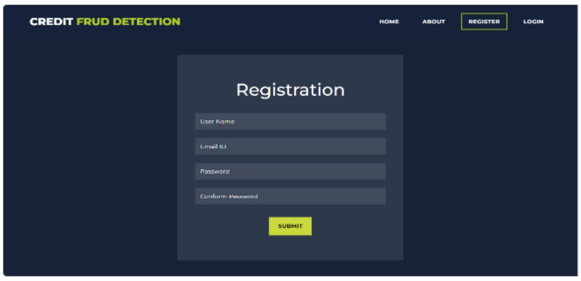
Optimization techniques such as grid search or random search are employed to systematically explore different hyper parameter combinations and identify the optimal settings. Cross-validation is used to evaluate model performance and ensure that it generalizes well to unseen data. This comprehensive training and tuning process target to boost the overall performance of the fraud detection system, resulting in a more precise and trustworthy identification of fraudulent transactions.



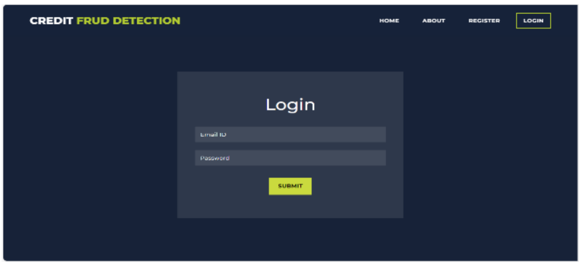
architecture diagram

OUTPUT SCREEN :

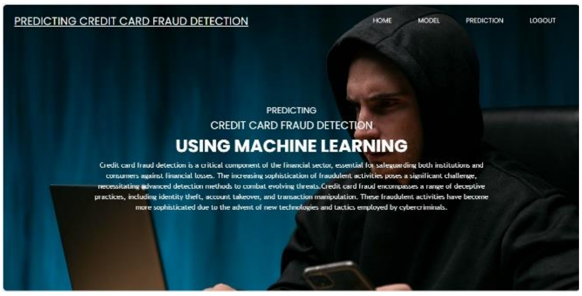
Registration Page:



Login Page:

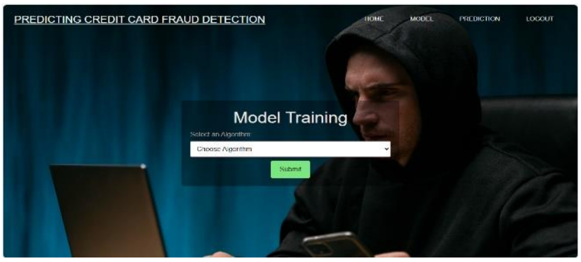


Model Home Page:


PREDICTING CREDIT CARD FRAUD DETECTION
HOME
MODEL
PREDICTION
LOGOUT
PREDICTING
CREDIT CARD FRAUD DETECTION
USING MACHINE LEARNING

Credit card fraud detection is a critical component of the financial sector, essential for safeguarding both institutions and consumers against financial losses. The increasing sophistication of fraudulent activities poses a significant challenge, necessitating advanced detection methods to combat evolving threats.Credit card fraud encompasses a range of deceptive practices, including identity theft, account takeover, and transaction manipulation. These fraudulent activities have become more sophisticated due to the advent of new technologies and tactics employed by cybercriminals.

Model Selection page:



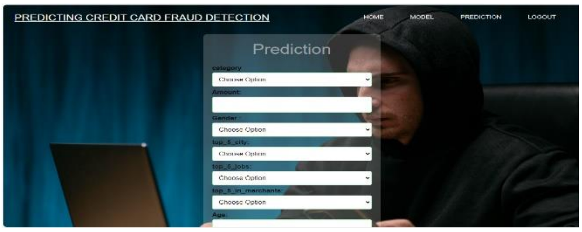
PREDICTING CREDIT CARD FRAUD DETECTION

Model Training

Select an Algorithm:

Choose Algorithm

Prediction page:



PREDICTING CREDIT CARD FRAUD DETECTION

Prediction

category:
Choose Option

Amount:

Gender :
Choose Option

top_5_city:
Choose Option

top_5_jobs:
Choose Option

top_5_in_marchants:
Choose Option

Age:

6. CONCLUSION

The intention of this study was to compare four ml algorithms for fraud detection. Findings indicate that each of the four algorithms, namely CNN, LSTM, Decision Trees, and Random Forests, demonstrated promising performance in identifying fraudulent transactions.

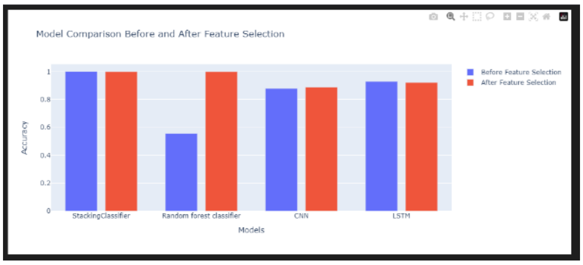


Figure Accuracy comparison

Key Results:

CNNs achieved an accuracy of 88.6% in detecting fraud, outperforming the other algorithms. This suggests that CNNs are effective in capturing intricate patterns in transaction data.

LSTM showed an accuracy of 92.2%, indicating their ability to model sequential dependencies and temporal dynamics in transaction data.

7. REFERENCES

1. Adi Saputra1, Suharjito2L: Fraud Detection using Machine Learning in e-Commerce, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 9, 2019.

2. Dart Consulting, Growth of Internet Users in India and Impact on Country's Economy: https://www.dartconsulting.co.in/marketnews/growth-of-internet-users-in-india-and-impact-on-countryseconomy/

3. Ganga Rama Koteswara Rao and R.Satya Prasad, " -Shielding The Networks Depending On Linux Servers Against Arp Spoofing, International Journal of Engineering and Technology(UAE),Vol. 7, PP.75-79, May 2018, ISSN No: 2227-524X, DOI - 10.14419/ijet.v7i2.32.13531.

4. Heta Naik , Prashasti Kanikar: Detctionsbased on Machine Learning Algorithms,International Journal of Computer Applications (0975 – 8887) Volume 182 – No. 44, March 2019.

5. Navanshu Khare ,Saad Yunus Sait: DetctionsUsing Machine Learning Models and Collating Machine Learning Models, International Journal of Pure and Applied Mathematics Volume 118 No. 20 2018, 825-838 ISSN: 1314-3395.

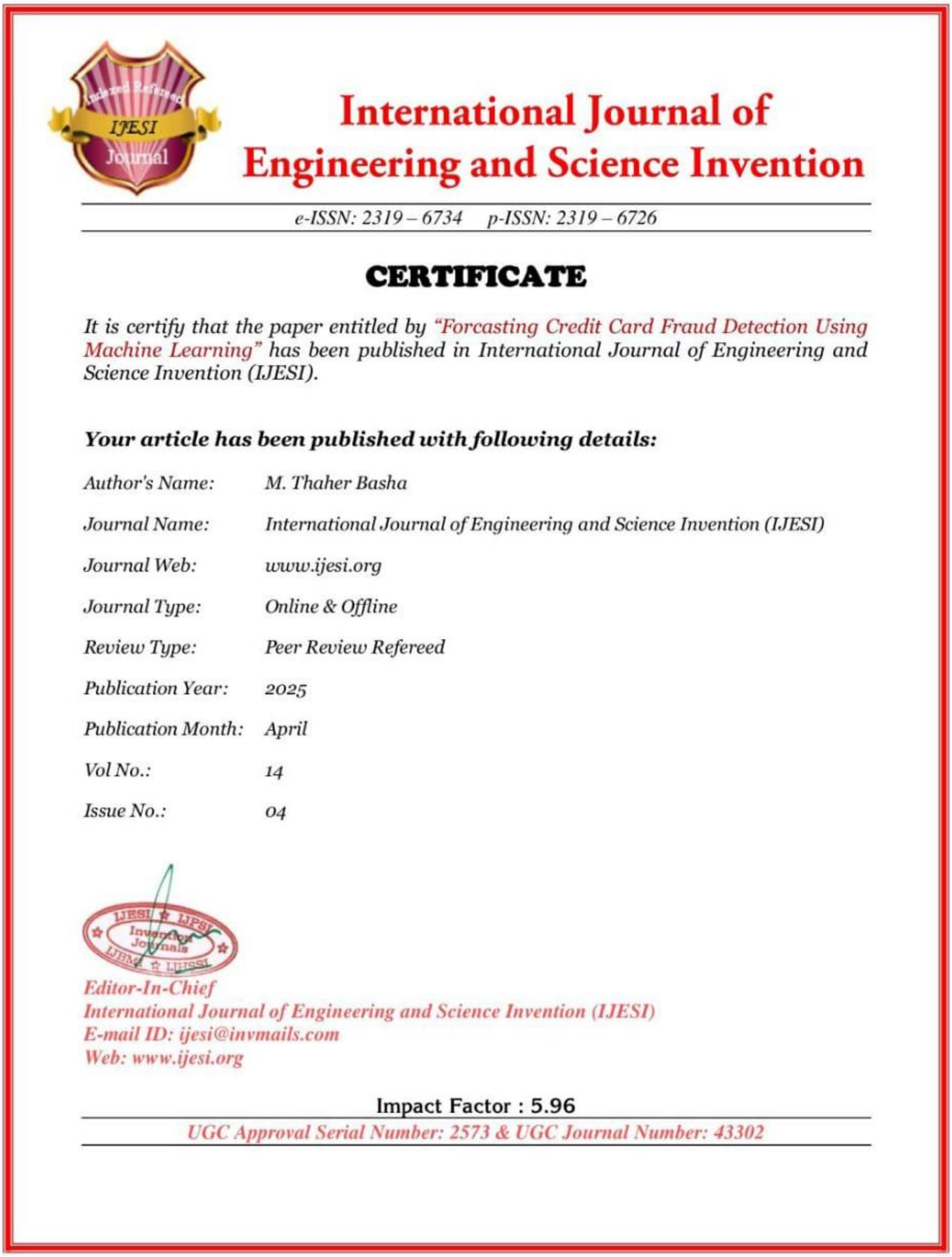
6. Randula Koralage, , Faculty of Information Technology, University of Moratuwa,Data Mining Techniques for Credit Card Fraud Detection.

7. Roy, Abhimanyu, et al:Deep learning detecting fraud in credit card transactions, 2018 Systems and Information Engineering Design Symposium (SIEDS), IEEE, 2018.

8.Sahayasakila.V, D. Kavya Monisha, Aishwarya, Sikhakolli VenkatavisalakshiseshsaiYasaswi: DetctionsSystem using Smote Technique and Whale Optimization Algorithm,International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019.

9. Statista.com. retail e-commerce revenue forecast from 2017 to 2023 (in billion U.S. dollars). Retrieved April 2020, from India : https://www.statista.com/statistics/280925/e-commercerevenueforecast-in-india/

10. Yashvi Jain, NamrataTiwari, Shripri yaDubey, Sarika Jain:A Comparative Analysis of Various DetctionsTechniques, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-5S2, January 2019.



International Journal of Engineering and Science Invention

e-ISSN: 2319 – 6734 p-ISSN: 2319 – 6726

CERTIFICATE

It is certify that the paper entitled by "Forcasting Credit Card Fraud Detection Using Machine Learning" has been published in International Journal of Engineering and Science Invention (IJESI).

Your article has been published with following details:

Author's Name:	*M. Thaher Basha*
Journal Name:	*International Journal of Engineering and Science Invention (IJESI)*
Journal Web:	*www.ijesi.org*
Journal Type:	*Online & Offline*
Review Type:	*Peer Review Refereed*
Publication Year:	*2025*
Publication Month:	*April*
Vol No.:	*14*
Issue No.:	*04*

IJESI ★ IJPSI Invention Journals IJBMI ★ IJHSSI

Editor-In-Chief
International Journal of Engineering and Science Invention (IJESI)
E-mail ID: ijesi@invmails.com
Web: www.ijesi.org

Impact Factor : 5.96

UGC Approval Serial Number: 2573 & UGC Journal Number: 43302