CRIME DETECTION IN CREDIT CARD FRAUD

A PROJECT REPORT

Submitted by

ELAKKIYA M (8115U23EC025)

in partial fulfillment of requirements for the award of the course

EGB1201 - JAVA PROGRAMMING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

K. RAMAKRISHNAN COLLEGE OF ENGINEERING
(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

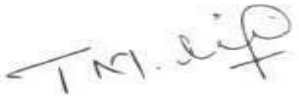
DECEMBER - 2024

K. RAMAKRISHNAN COLLEGE OF ENGINEERING (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"CRIME DETECTION IN CREDIT CARD FRAUD"** is the bonafide work of **ELAKKIYA M (8115U23EC025)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



SIGNATURE

Dr. T. M. NITHYA, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of CSE

K.Ramakrishnan College of Engineering (Autonomous)

Samayapuram–621112.

SIGNATURE

Mr.V.KUMARARAJA, M.E.,(Ph.D.,),

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Engineering (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 06/12/24

INTERNAL EXAMINER

EXTERNAL EXAMINER

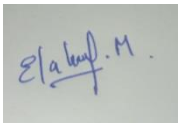
DECLARATION

I declare that the project report on **"CRIME DETECTION IN CREDIT CARD FRAUD"** is the result of original work done by us and best of our knowledge, similar workhas not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **EGB1201 -**

JAVA PROGRAMMING.

.

Signature



ELAKKIYA M

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Engineering (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequateduration in completing our project.

I would like to thank **Dr. D. SRINIVASAN, B.E, M.E., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. M. NITHYA, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Mr. V. KUMARARAJA, M.E., (Ph.D.,),** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patiencewhich motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To achieve a prominent position among the top technical institutions.

MISSION OF THE INSTITUTION

- M1: To bestow standard technical education par excellence through state of the art infrastructure, competent faculty and high ethical standards.
- M2: To nurture research and entrepreneurial skills among students in cutting edge technologies.
- M3: To provide education for developing high-quality professionals to transform the society.

VISION OF DEPARTMENT

To create eminent professionals of Computer Science and Engineering by imparting quality education.

MISSION OF DEPARTMENT

M1: To provide technical exposure in the field of Computer Science and Engineering through state of the art infrastructure and ethical standards.

M2: To engage the students in research and development activities in the field of Computer

Science and Engineering.

M3: To empower the learners to involve in industrial and multi-disciplinary projects for addressing the societal needs.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1: Analyse, design and create innovative products for addressing social needs.

PEO2: Equip themselves for employability, higher studies and research.

PEO3: Nurture the leadership qualities and entrepreneurial skills for their successfulcareer.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO1:** Apply the basic and advanced knowledge in developing software, hardware and firmware solutions addressing real life problems.

- **PSO2:** Design, develop, test and implement product-based solutions for their career enhancement.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

Credit card fraud has become a significant concern in the digital era, as the increasing reliance on electronic payments has created new opportunities for fraudulent activities. Effective crime detection in credit card fraud is essential to safeguard financial institutions and customers from financial losses and breaches of trust. This study explores advanced methods and technologies employed in detecting credit card fraud, focusing on machine learning, artificial intelligence, and big data analytics. These tools analyze vast datasets to identify patterns, anomalies, and suspicious activities in real time, enabling early intervention and fraud prevention. The integration of predictive modeling and behavioral analysis enhances the precision of fraud detection, ensuring greater accuracy in distinguishing legitimate transactions from fraudulent ones.

This research also examines the challenges associated with implementing fraud detectionsystems,suchasmaintaininguserprivacy,balancingfalsepositivesandnegatives, and adapting to evolving fraud techniques. By leveraging dynamic algorithms and continuous learning models, financial institutions can stay ahead of increasingly sophisticated fraudsters. The findings underline the importance of collaboration between technologyproviders,financialinstitutions,andregulatoryauthoritiestocreateasecureand trustworthypaymentecosystem.Thestudyconcludeswithrecommendationsforenhancing frauddetectionstrategiestoprotectconsumersandensuretheintegrityofdigitaltransactions.

.

ABSTRACTWITHPOsANDPSOsMAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
Credit card fraud detection has become a critical challenge in the era ofdigital transactions. This study focuses on leveraging advanced techniques such as machine learning and data analytics to identify fraudulentactivitieswith highaccuracy.Byanalyzing transaction patterns, behavioral anomalies, and real-time data, these systems enhance the detection of unauthorized transactions while minimizing false positives. The research highlights the importance of adaptive algorithms to counter evolving fraud techniquesandemphasizestheneedforacollaborative approach among financial institutions and technology providers to create a secure payment environment	**PO1** **PO2** **PO3** **PO5** **PO6** **PO7**	**PSO1** **PSO2**

SUPERVISOR **HEADOFTHEDEPARTMENT**

TABLEOFCONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective:

The objective of this project is to design and implement anefficient credit card fraud detection system using Java. The system will leverage Java's robust features to process and analyze transaction data in real time, identifying fraudulent activities through algorithms and machine learning models. By utilizing Java's libraries and frameworks,suchasWekafordata miningandApache Sparkforbigdataprocessing, thesolutionaimstoensurehighaccuracyandscalability. Furthermore,thesystemwill focus oncreatinguser-friendlyinterfaces andsecure data handling mechanisms while minimizing false positives, thereby enhancing fraud detection capabilities and maintaining trust in digital payment systems.

1.2 Overview:

The project leverages Java-based frameworks like Apache Spark for big data processingandWekaforintegratingmachinelearningalgorithms.Thesetoolsenable thedetectionofanomaliesbyanalyzingtransactionhistories,userbehaviors,andrisk patterns.Byimplementingsecure codingpractices andJava'sinbuiltdata encryption libraries, the system ensures secure handling of sensitive financial information. The use of JavaFX for building intuitive user interfaces further enhances the system's usability. Overall, Java serves as a versatile platform to deliver a reliable, accurate, and user-friendly credit card fraud detection solution.

1.3 JavaProgrammingConcepts

Encapsulation: Protects sensitive data such as user information and transaction details by using private fields and providing controlled access through getters and setters.

Inheritance: Allows creating a hierarchy of fraud detection modules by extending base classes, e.g., a FraudDetection base class extended by specific fraud strategies like CreditCardFraud or IdentityTheftDetection.

Polymorphism: Enables the system to handle different fraud detection techniques dynamically (e.g., machine learning-based, rule-based).

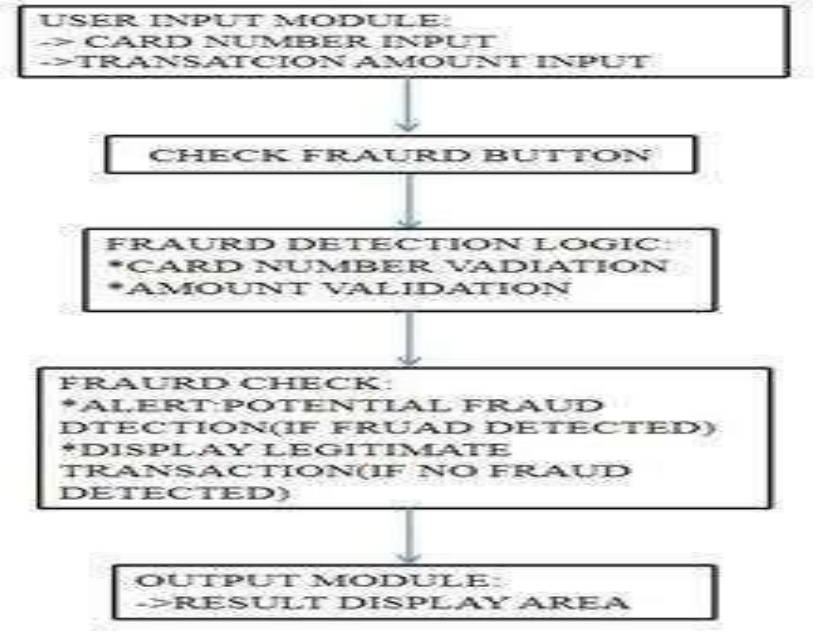
CHAPTER 2

PROJECTMETHODOLOGY

2.1 ProposedWork:

Theproposedworkaimstodevelopanintelligent CreditCardFraudDetection System using Java, integrating key components like transaction monitoring, risk assessment, fraud detection, alert management, and execution management. The system will be designed to detect and prevent fraudulent activities in real-time by continuously monitoring credit card transactions for suspicious patterns. It will leverage machine learning algorithms and rule-based systems to assess transaction risks and classify them as either legitimate or fraudulent. Suspicious transactions will trigger immediate alerts, notifying users or administrators to take further actions.

2.2 BlockDiagram



CHAPTER 3

MODULEDESCRIPTION

3.1 TRANSACTIONMONITERING

Transaction monitoring in a credit card fraud detection system involves continuously tracking and analyzing transactions in real-time to identify any suspicious behavior. By evaluating factors like transaction amount, merchant type, geographical location, andfrequency,thesystemcanflaganomaliesthatdeviatefromauser'stypical behavior. In Java, this can be achieved by using event listeners or scheduled tasks to compare incoming transactions with historical data and predefined patterns. Real-time monitoring allows the system to respond immediately to potential fraud, ensuring that suspicious transactions are quickly detected and acted upon, such as blocking further transactions or sending alerts.

3.2 FRUADDETECTION

Risk assessment in fraud detection evaluates the likelihood that a transaction is fraudulent by assigning a risk score based on factors like transaction value, location, and timing. Java can implement scoring systems that categorize transactionsintolow,medium,orhighrisk,wherehigh-risktransactions,suchasthose withlargeamountsormadeinunusuallocations, areflaggedforfurtherreview. Using machine learning techniques, such as decision trees or logistic regression, the system can automatically adjust risk scores based on patterns learned from past transactions.

Thisdynamicassessmentallowsforaccurateidentificationofpotentiallyfraudulent activity, enabling timely interventions.

3.3 RISKASSESSMENT

Fraud detection focuses on identifying transactions that exhibit patterns or characteristicsassociatedwithfraudulentactivities.InaJava-basedsystem,fraud detection can be implemented using rule-based methods or machine learning algorithms to flag transactions that deviate from normal behavior, such as large purchases inunfamiliar locations. Javaframeworks likeWekaorDeeplearning4j can be used to apply machine learning models, which learn from historical transaction data to identify and classify potentially fraudulent transactions. This approach ensures that new fraud patterns, even those previously unseen, can be detected efficiently.

3.4 ALERTMANAGEMENT

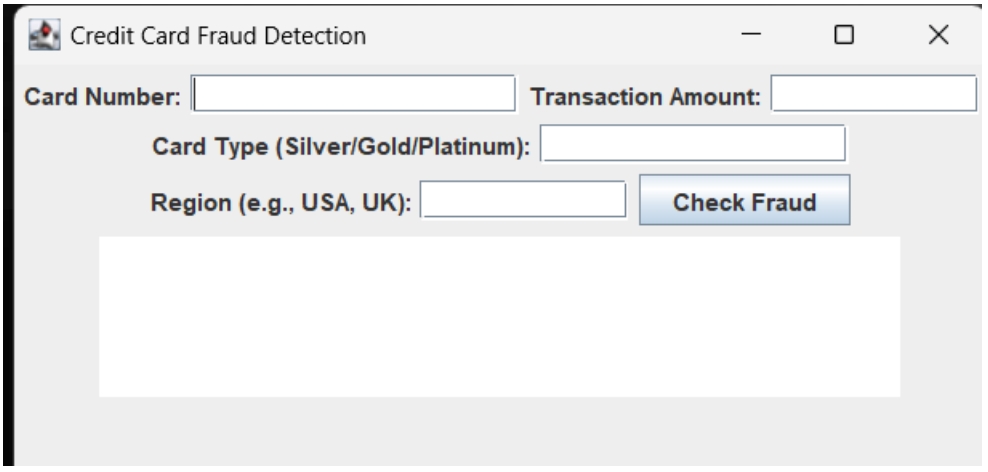
Alert management involves notifying the appropriate stakeholders when a suspicioustransactionisdetected.InJava,alertscanbegeneratedviaemail,SMS, or application-based notifications once a high-risk or fraudulent transaction is flagged. The system can automatically notify the cardholder for verification or alertthefrauddetectionteamforfurtherinvestigation.JavalibrarieslikeJavaMail API or Twilio can be used to send these alerts, ensuring that timely actions are taken to prevent further fraud. The system can also store alerts in a database for tracking and escalation, allowing administrators to review and respond based on the severity of the detected fraud.

3.5 EXECUTIONMANAGEMENT:

Execution management refers to the automated actions taken after fraud is detected, suchas freezinga credit card or blockinga transaction. InJava, this canbe achievedby integratingthe frauddetectionsystemwithexternalsystems, suchas the bank's transaction processing infrastructure. Once fraud is detected, the system can initiate actions like blocking the card or stopping further transactions through REST APIsordirectintegrationwithbankingsystems.Executionmanagementalsoinvolves logging all actions for audit and compliance purposes, with Java's logging frameworks like Log4j capturing the sequence of fraud-related actions, ensuring transparency and accountability in the response process.

CHAPTER 4

RESULTSANDDISCUSSION



DISSCUSION:

The Credit Card Fraud Detection Program demonstrates key concepts of modular programming and user interaction. The UI Components Initialization module provides a simple interface for input and result display. The Action Listener handles user interactions, validating inputs for correctness. The Fraud Detection Logic serves as the core, using basic rules to identify potentially fraudulent transactions. The Window Closing Eventen sures smooth termination of the application. Finally, the Main Method integrates all components, providing a seamless user experience. While basic, the program emphasizes modularity and introduces foundational concepts for fraud detection system

CHAPTER 5

CONCLUSION

The developmentofa credit card fraud detection systemrelies onthe seamless integrationofmultiplemodulesdesignedtoensuresecurity,real-timeprocessing,and accurate frauddetection. TheUserManagementModuleensuresthatonlyauthorized users can access the system, safeguarding sensitive personal and financial information. The Transaction Processing Module validates and records every transaction, forming the backbone for detecting suspicious activities.

The Fraud Detection Module is at the heart of the system, using advanced algorithms, machine learning models, and rule-based checks to identify potentially fraudulent transactions with minimal false positives. Finally, the Notification and Reporting Module ensures that users and administrators are promptly alerted to any suspicious activity and have access to detailed reports for further investigation.Together, these modules provide a robust framework for preventing fraud, reducing financial losses, and maintaining the integrity of credit card transactions,makingthesystemessentialforbothusersandfinancialinstitutions.By continuously improving the fraud detection algorithms and ensuringscalability, this system can effectively combat emerging fraud trends, offering enhanced security in the digital payment ecosystem.

REFERENCES

1. Books:

shabh, K., & Gupta, D. (2019). "Credit Card Fraud Detection Using Machine Learning Algorithms: A Comparative Analysis." Journal of Computer Science and Technology, 34(3), 547-559.YouTube Channels:

1. ProgrammingwithMosh
2. Telusko
3. TheCodingTrain

2. AdditionalResources:

Zhang, Y., & Ma, Z. (2018). "Credit Card Fraud Detection Based on EnsembleLearningandFeatureEngineering."Computers,Materials&Continua, 58(2), 53

APPENDIX A (SOURCE CODE)

```
import javax.swing.*; import
java.awt.*;
importjava.awt.event.*;
import java.util.*;


publicclassCreditCardFraudDetectionextendsJFrameimplementsActionListener{
   privateJTextFieldcardNumberField;

   private JTextFieldtransactionAmountField;
   private JTextFieldcardTypeField;

   private JTextFieldregionField;
   private JTextArearesultArea;
   privateJButtoncheckFraudButton;


   privateMap<String,java.util.List<Transaction>>transactionHistory=newHashMap<>(); private
   static final double HIGH_AMOUNT_THRESHOLD = 5000.0;

   privatestaticfinalintMAX_TRANSACTIONS_PER_HOUR=3;

   public CreditCardFraudDetection() {
      setTitle("CreditCardFraudDetection");
      setSize(500, 400);

      setLayout(newFlowLayout());


      //InitializeUIcomponents

      JLabelcardNumberLabel=newJLabel("CardNumber:");
      cardNumberField = new JTextField(16);

      JLabeltransactionAmountLabel=newJLabel("TransactionAmount:");
      transactionAmountField = new JTextField(10);

      JLabelcardTypeLabel=newJLabel("CardType(Silver/Gold/Platinum):"); cardTypeField
      = new JTextField(15);


      JLabelregionLabel=newJLabel("Region(e.g.,USA,UK):"); regionField =
      new JTextField(10);

      checkFraudButton=newJButton("CheckFraud");
      resultArea = new JTextArea(5, 40);
      resultArea.setEditable(false);
```

```
    //Addcomponentstotheframe
    add(cardNumberLabel);
    add(cardNumberField);
    add(transactionAmountLabel);
    add(transactionAmountField);
    add(cardTypeLabel);
    add(cardTypeField);
    add(regionLabel);


    add(regionField);
    add(checkFraudButton);
    add(resultArea);

    // Add action listener for the button
    checkFraudButton.addActionListener(this);

    //Setframevisibility
    setVisible(true);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

@Override

publicvoidactionPerformed(ActionEvente){ if
    (e.getSource() == checkFraudButton) {

        String cardNumber = cardNumberField.getText();
        String amountStr = transactionAmountField.getText();
        String cardType = cardTypeField.getText().trim();
        String region = regionField.getText().trim();

        try{

            doubleamount=Double.parseDouble(amountStr);

            Stringresult=checkForFraud(cardNumber,cardType,amount,region);
            resultArea.setText(result);

        }catch(NumberFormatExceptionex){

            resultArea.setText("Invalidtransactionamount.Pleaseenteravalidnumber.");

        }

    }
}

privateStringcheckForFraud(StringcardNumber,StringcardType,doubleamount,Stringregion){

    //Getthetransactionhistoryforthecard

    java.util.List<Transaction>transactions=transactionHistory.getOrDefault(cardNumber,new ArrayList<>());
```

```
//Checkforfraudbasedondifferentcriteria
if (amount>HIGH_AMOUNT_THRESHOLD){
   return"Potentialfrauddetected:Transactionamountexceedsthreshold.";
}

//Checkformultipletransactionswithinthelasthour long
currentTime = System.currentTimeMillis();
longoneHourAgo=currentTime-3600000;//1hourinmilliseconds
longrecentTransactions=transactions.stream().filter(t->t.timestamp>oneHourAgo).count();

if (recentTransactions>= MAX_TRANSACTIONS_PER_HOUR)
   {return"Potentialfrauddetected:Multipletransactionsinashortperiod.";
}
//Addthenewtransactiontothehistory
TransactionnewTransaction=newTransaction(amount,currentTime,region);
```

```
        transactions.add(newTransaction);
        transactionHistory.put(cardNumber,
        transactions);

        //Checkforcardtype-
        specifictransactionlimits double limit =
        getTransactionLimit(cardType); if (limit
        == -1) {

            return"Invalidcardtype.";

        }

        if(amount>limit){

            return"Potentialfrauddetected:Transactionexceedslimitfor"+cardType+"card.";

        }

        return"Transactionislikelylegitimate.";

    }

    //Gettransactionlimit basedoncardtype

    privatedoublegetTransactionLimit(StringcardType)
        { switch (cardType.toLowerCase()) {

            case"silver":retu
            rn1000; case
            "gold": return
            5000;

            case "platinum": return
            10000; default:return-
            1;//Invalidcardtype

        }

    }

    publicstaticvoidmain(String[]args
        ){ new
        CreditCardFraudDetection();

    }

    //Innerclasstorepresentatran
    saction class Transaction {
        Transaction(doubleamount,longtimestamp,Stringre
            gion){ this.amount = amount;

            this.timestamp=timest
            amp; this.region =
            region;

        }

    }

}
```