CREDIT CARD FRAUD DETECTION

Enroll no. - 101303

Name - Jagmann Singh Grewal

Supervisor - Prof. Dr. Nitin

May, 2014

Submitted in partial fulfillment of Degree of Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT.

Table of Contents

CERTIFICATE

This is to certify that the work entitled-**"CREDIT CARD FRAUD DETECTION"** submitted by **JAGMANN SINGH GREWAL-101303** in partial fulfillment for award for degree of Bachelor of Technology in Information Technology of JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY has been carried out under my supervision. This work has not been submitted partially or wholly to any other University for any award of this or any other degree.

Prof. Dr. Nitin
Department of Computer Science Engineering and Information Technology
Jaypee University of Information Technology
Waknaghat

Acknowledgement

"It is not possible to prepare a project without the assistance & Encouragement of other people. This one is certainly no exception."

On the very outset of this report, we would like to extend our sincere & heartfelt obligation towards all the personages who have helped us in this endeavor. Without their active guidance, help, cooperation & encouragement, we would not have made headway in the project.

We would like to show our greatest appreciation to **Prof. Dr. Nitin** . We feel motivated every time we get his encouragement. For his coherent guidance throughout the tenure of the project, we feel fortunate to be taught by **Prof. Dr. Nitin**, who gave us his unwavering support. Besides being our mentor, he taught us that there is no substitute for hard work.

We owe our heartiest thanks to Brig. (Retd.) S.P. Ghrera (HOD, CES/IT Department) who has always inspired us to take initiatives and showed us the path for achieving our goal.

JAGMANN SINGH GREWAL-101303

SYNOPSIS

The project titled **"CREDIT CARD FRAUD"** detects the fraudulent card during transactions and alerts the customer regarding the fraud. This project also aims in minimizing the number of false alerts. The concept of genetic algorithm is a novel one in this application domain.

The algorithm begins with multi-population of randomly generated chromosomes. These chromosomes undergo the operations of selection, crossover and mutation. Crossover combines the information from two parent chromosomes to produce new individuals, exploiting the best of the current generation, while mutation or randomly changing some of the parameters allows exploration into other regions of the solution space. Natural selection via a problem specific cost function assures that only the best fit chromosomes remain in the population to mate and produce the next generation. Upon iteration, the genetic algorithm converges to a global solution.

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

In recent years, the prevailing data mining concerns people with credit card fraud detection model based on data mining. Since our problem is approached as a classification problem, classical data mining algorithms are not directly applicable. So an alternative approach is made by using general purpose meta heuristic approaches like genetic algorithms.

This project is to propose a credit card fraud detection system using genetic algorithm. Genetic algorithms are evolutionary algorithms which aim at obtaining better solutions as time progresses. When a card is copied or stolen or lost and captured by fraudsters it is usually used until its available limit is depleted. Thus, rather than the number of correctly classified transactions, a solution which minimizes the total available limit on cards subject to fraud is more prominent. It aims in minimizing the false alerts using genetic algorithm where a set of interval valued parameters are optimized.

1.2 PLAN OF THE REPORT

It is documented in such way that, it is convenient to the user. Each section is divided into sub-sections. The introduction of each section is provided below:

Chapter 2: Problem Definition and Feasibility Analysis

This chapter gives the information regarding analysis done for the proposed system. Here the goal of the project is explained, and also the cost and performance factors which will affect the feasibility of the project is explained

Chapter 3: Software Requirements Specification

This chapter gets through the functional and non functional requirement phase of the proposed system. This chapter illustrates the overall structure and responsibility of the project using UML.

Chapter 4: System Analysis

This chapter gets through the requirement phase of the proposed system and studies the requirements of the system in detail. It presents a formal document that crystallizes the user's requirements. The result of this study is being used in all the future steps of development of the project.

Chapter 5: System Design

In this chapter the detailed system design explores architecture of the system. It deals with the modules and their relationship in building the whole system. Design at this level explains about sub systems which are building blocks of the whole system. These sub systems have their well defined functionality.

Chapter 6: Coding, Testing and Implementation

The coding logic of the tools is explained with the code and syntax. We present how the code is organized with comments on code for understanding in future reference. We also discuss the Naming conventions that were followed during the Implementation phase of the project and also the descriptions of the methods of all the modules used by the system.

Chapter 7: Conclusion and Foreseeable Enhancements

This chapter gives the conclusion of the report and also the possible enhancements that could be done in the future.

CHAPTER 2

PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

2.1 PROBLEM DEFINITION

To develop a credit card fraud detection system using genetic algorithm. During the credit card transaction, the fraud is detected and the number of false alert is being minimized by using genetic algorithm. Instead of maximizing the numbers of correctly classified transactions we defined an objective function where the misclassification costs are variable and thus, correct classification of some transactions are more important than correctly classifying the others.

The algorithm begins with multi-population of randomly generated chromosomes. These chromosomes undergo the operations of selection, crossover and mutation. Crossover combines the information from two parent chromosomes to produce new individuals, exploiting the best of the current generation, while mutation or randomly changing some of the parameters allows exploration into other regions of the solution space. Natural selection via a problem specific cost function assures that only the best fit chromosomes remain in the population to mate and produce the next generation. Upon iteration, the genetic algorithm converges to a global solution.

2.1.1 LITERATURE SURVEY

Fraud detection has been usually seen as a data mining problem where the objective is to correctly classify the transactions as legitimate or fraudulent. For classification problems many performance measures are defined most of which are related with correct number of cases classified correctly.

A more appropriate measure is needed due to the inherent structure of credit card transactions. When a card is copied or stolen or lost and captured by fraudsters it is usually used until its available limit is depleted. Thus, rather than the number of correctly classified transactions, a solution which minimizes the total available limit on cards subject to fraud is more prominent.

Since the fraud detection problem has mostly been defined as a classification problem, in addition to some statistical approaches many data mining algorithms have been proposed to solve it. Among these, decision trees and artificial neural networks are the most popular ones. The study of Bolton and Hand provides a good summary of literature on fraud detection problems.

However, when the problem is approached as a classification problem with variable misclassification costs as discussed above, the classical data mining algorithms are not directly applicable; either some modifications should be made on them or new algorithms developed specifically for this purpose are needed. An alternative approach could be trying to make use of general purpose meta heuristic approaches like genetic algorithms.

Genetic algorithm

Genetic algorithms are evolutionary algorithms which aim at obtaining better solutions as time progresses. Since their first introduction by Holland, they have been successfully applied to many problem domains from astronomy to sports, from optimization to computer science, etc. They have also been used in data mining mainly for variable selection and are mostly coupled with other data mining algorithms. In this study, we try to solve our classification problem by using only a genetic algorithm solution.

Pseudo code of genetic algorithm

Initialize the population

Evaluate initial population

Repeat

Perform competitive selection

Apply genetic operators to generate new solutions

Evaluate solutions in the population

Until some convergence criteria is satisfied.

Selection process

Selection is used for choosing the best individuals, that is, for selecting those chromosomes with higher fitness values. The selection operation takes the current population

and produces a ‘mating pool’ which contains the individuals which are going to reproduce. There are several selection methods, like biased selection, random selection, roulette wheel selection, tournament selection. In this work the following selection mechanisms are used.

Tournament Selection

Tournament selection has been used in this as it selects optimal individuals from diverse groups. It selects t individuals from the current population uniformly at random, forms a tournament and the best individual of a group wins the tournament and is put into the mating pool for recombination. This process is repeated the number of times necessary to achieve the desired size of intermediate population. The tournament size controls the selection strength. The larger the tournament size, the stronger is the selection process.

Elitist Selection

In order to make sure that the best individuals of the solution are passed to further generations, and should not be lost in random selection, this selection operator is used. So we used a few best chromosomes from each generation, based on the higher fitness value and are passed to the next generation of population.

Reproduction

To generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", some research suggests more than two "parents" are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

Although Crossover and Mutation are known as the main genetic operators, it is possible to use other operators such as regrouping, colonization-extinction, or migration in genetic algorithms.

Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Combinations of the above

2.2 FEASIBILITY ANALYSIS:

A feasibility analysis is an important tool to help you assess the viability of starting a new value-added business, or re-organizing or expanding an existing business.

All projects are feasible provided with unlimited resources and infinite time. But unfortunately, scarcity of resources and difficult delivery dates plagues all projects.

The following three kinds of feasibilities are studied in the feasibility analysis of the project.

- Operational feasibility

- Technical feasibility
- Economical feasibility.

2.2.1. OPERATIONAL FEASIBILITY

The operational scope of the system is verified under operational feasibility. The proposed system will have enough operational reach, which ensures the security of the information. Hence, operational feasibility of the proposed system is found to be high.

This project involves the general user friendly windows environment. Graphical user Interface, being today de fecto standard, has been exploited to give the user a nice look and feel. Operational feasibility ensures that the project is successfully implemented. The project can be used by the users with basic internet knowledge. Hence we conclude that this project is operationally feasible.

2.2.2. TECHNICAL FEASIBILITY

Technical feasibility checks the technical possibilities of the system to be developed. Necessary hardware and software resources to develop the system are readily available. Hence, the technical feasibility of the system is more. This is the study where the technical requirements of the proposed system are checked and the efficiency of the newly developed project to work in the existing technical requirements of the system is also checked. Information regarding the upgrades in the technical aspects is gathered and is estimated with the technical features of the existing system. If the technical features that are available in the existing system are suited to accommodate the proposed system, then the system that has been developed is said to be technically feasible. As all the technology for this project is available in the latest Browsers, this project is technically feasible.

2.2.3. ECONOMICAL FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly it is known as cost/benefit analysis. The software used in this project is freeware so the cost of developing the tool is minimal. It requires very easy technique and minimal software. So it does not need much cost and software. So, it can be used in any environment.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1. INTRODUCTION.

SRS is basically an organization understanding of a customer or potential clients system and dependencies at a particular point in time prior to any actual design or development work. Software requirement specification has been developed for future reference in case of any ambiguity and misunderstanding. SRS provides a detailed of the requirements, behaviors', constraints and performance of the system.

3.2. REQUIREMENT ANALYSIS

Requirement analysis is for transformation of operational need into software description, software performance parameter, and software configuration through use of standard, iterative process of analysis and trade-off studies for understanding what the customer wants analyzing need, assessing feasibility, negotiating a reasonable solution validating the specification and managing the requirements.

3.2.1. PURPOSE

The purpose of this document is to define the requirements of credit card fraud detection. In detail, this document will provide a general description of our project, including user requirements, product perspective, and overview of requirements, general constraints. In addition, it will also provide the specific requirements and functionality needed for this project - such as interface, functional requirements and performance requirements.

3.2.2. SCOPE

The scope of this SRS document persists for the entire life cycle of the project. This document defines the final state of the software requirements agreed upon by the customers and designers. Finally at the end of the project execution all the functionalities may be traceable from the SRS to the product. The document describes the functionality, performance, constraints, interface and reliability for the entire life cycle of the project.

3.2.3. OVERVIEW

The software requirement specification document for the system covers the following two sections:

GENERAL DESCRIPTIONS:

It provides the general description about the project. It includes description about the product function, user characteristics and general constraint.

SPECIFICATION REQUIREMENT:

This section describes about both the functional and non functional requirement of the system. The functional requirement section defines the system external interface, general requirement, performance, design constraint etc.

3.2.4. GENERAL DESCRIPTIONS

The credit card fraud detection system has been developed to alert the customer regarding the fraud of their credit card. After the payment process the transactions performed is verified whether the performed transaction is real or fraud transaction and minimizes the false alert by implementing genetic algorithm.

3.2.4.1. PRODUCT FUNCTION

The project is guaranteed to provide reliable results and the functionality of the product to detect the fraud transactions effectively and provide flexibility to the user in a secured and accurate manner.

3.2.4.2. USER CHARACTERISTICS

The user of the system are classified as customers and administrator,

- Customers are those who make the transaction through any means.
- Administrator who computes on the transaction and reports about the fraud usage

3.2.4.3. GENERAL CONSTRAINTS

- **Hardware Limitations:** There are no hardware limitations.
- **Interfaces to other Applications:** There shall be no interfaces.

- **Parallel Operations:** There are parallel operations.
- **Audit Functions:** There shall be no audit functions.
- **Control Functions:** There shall be no control functions

3.2.5. FUNCTIONAL REQUIREMENTS

The relationship between the input and output to the system is determined by the functional requirement of the SRS.

3.2.5.1 TECHNICAL ISSUES

Many a software project has failed due to an incomplete or inaccurate analysis process, especially technical issues. Technical issues are a key step while developing a software application.

3.2.5.2 RISK ANALYSIS

Project Risk Analysis is for Cost estimates of known accuracy and risk on capital investment projects. Their main challenge is to determine how to model and visualize the complex relationships between risks, define and monitor the risks' impacts, analyze the probability of risk occurrence, mitigate the negative impact of risks, and monitor the course of the project with risks and uncertainties.

3.2.6. INTERFACE REQUIREMENTS

The system performance is adequate. However, Virtual travel agency is working with the user internet connection, 60% of the performance is up to the client side.

3.2.6.1 HARDWARE REQUIREMENTS

- Processor type : Pentium III-compatible processor or faster.
  Processor speed : Minimum: 1.0 GHz, Recommended: 2.0 GHz or faster
- RAM : 512 MB or more
- HARD DISK : 20GB or more
- Monitor : VGA or higher resolution 800x600 or higher resolution
- Pointing device : Microsoft Mouse or compatible pointing device
- CD-ROM : Actual requirements will vary based on system configuration and the applications and features chosen to install.

3.2.6.2 SOFTWARE REQUIREMENTS

- Application software Framework : Java
- Back End : SQL Server
- Operating System : Windows XP Professional or more

3.2.7. PERFORMANCE REQUIREMENTS

- The project has the following performance requirements
- The prime requirement is that no error condition causes a project to exit abruptly.
- Any error occurred in any process should return an understandable error message
- The response should be fairly fast, the action participants should not be confused at any point of time about action that is happening.
- The system performance is adequate.

3.2.8 NON FUNCTIONAL ATTRIBUTES

3.2.8.1 SECURITY

The project provides a security to different kind of customers by means of authentication level. The authorization mechanism of the system will block the unwanted attempts to the server.

3.2.8.2RELIABILITY

The project is guaranteed to provide reliable results for the entire user. The system shall operate 95% of the time. The number of defect should not exceed 10 per function. In addition, before the submission of the final release the calendar must be tested in case of the defects over 10 per function.

3.2.8.3 USABILITY

- Since GUI interface is used, it can be used by a user.
- Since the system is placed on for online users any type user can use the system.
- The system detects the fraud and reports to the user.

3.2.8.4 SCALABILITY

The need for scalability has been a driver for much of the technology innovations of the past few years. The industry has developed new software languages, new design strategies, and new communication and data transfer protocols, in part to allow web sites to grow as needed.

3.2.8.5 MAINTAINABILITY

Maintainability is our ability to make changes to the product over time. We need strong maintainability in order to retain our early customers. We will address this by anticipating several types of change, and by carefully documenting our design and implementation.

CHAPTER 4

SYSTEM ANALYSIS

This chapter gives the information regarding analysis done for the proposed system. System Analysis is done to capture the requirement of the user of the proposed system. It also provides the information regarding the existing system and also the need for the proposed system. The key features of the proposed system and the requirement specifications of the proposed system are discussed below.

4.1 **EXISTING SYSTEM**

The Traditional detection method mainly depends on database system and the education of customers, which usually are delayed, inaccurate and not in-time. After that methods based on discrimate analysis and regression analysis are widely used which can detect fraud by credit rate for cardholders and credit card transaction. For a large amount of data it is not efficient.

4.2 PROBLEM RECOGNITION

The high amount of losses due to fraud and the awareness of the relation between loss and the available limit has to be reduced. The fraud has to be deducted in real time and the number of false alert has to be minimized.

4.3 PROPOSED SYSTEM

The proposed system overcomes the above mentioned issue in an efficient way. Using genetic algorithm the fraud is detected and the false alert is minimized and it produces an optimized result. The fraud is detected based on the customers behavior. A new classification problem which has a variable misclassification cost is introduced. Here the genetic algorithms is made where a set of interval valued parameters are optimized.

CHAPTER 5

SYSTEM DESIGN

The process of design involves "conceiving and planning out in mind and making a drawing, pattern or a sketch". The system design transforms a logical representation of what a given system is required to do into the physical reality during development. Important design factors such as reliability, response time, throughput of the system, maintainability, expandability etc., should be taken into account. Design constraints like cost, hardware limitations, standard compliance etc should also be dealt with. The task of system design is to take the description and associate with it a specific set of facilities-men, machines (computing and other), accommodation, etc., to provide complete specifications of a workable system.

This new system must provide for all of the essential data processing and it may also do some of those tasks identified during the work of analysis as optional extras. It must work within the imposed constraints and show improvement over the existing system.. At the outset of design a choice must be made between the main approaches. Talks of 'preliminary design" concerned with identification analysis and selections of the major design options are available for development and implementation of a system. These options are most readily distinguished in terms of the physical facilities to be used for the processing who or what does the work.

5.1 ARCHITECTURAL DESIGN

Describing the overall features of the software is concerned with defining the requirements and establishing the high level of the system. During architectural design, the various web pages and their interconnections are identified and designed. The major software components are identified and decomposed into processing modules and conceptual data structures and the interconnections among the modules are identified. The following modules are identified in the proposed system.

5.2 SYSTEM ARCHITECTURE

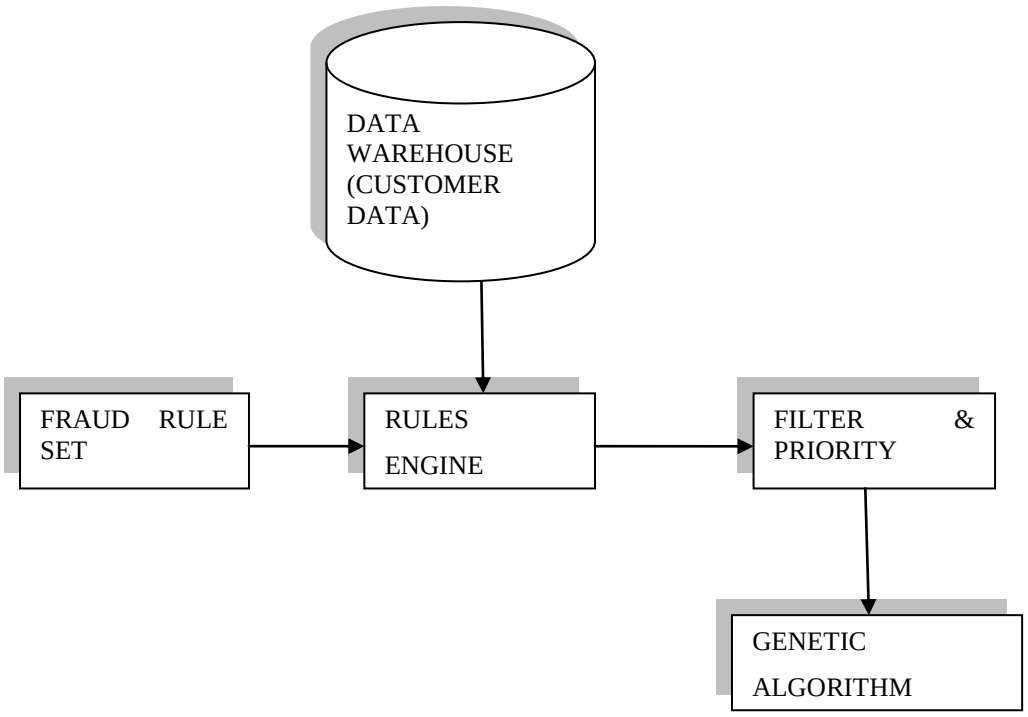


Figure 1 OVERALL SYSTEM DESIGN

The above architecture describes the work structure of the system.

- The customer data in the data warehouse is subjected to the rules engine which consists of the fraud rule set.
- The filter and priority module sets the priority for the data and then sends it to the genetic algorithm which performs its functions and generates the output.

5.3 DETAILED SYSTEM DESIGN

Detailed design deals with the various modules in detail explaining them with appropriate Diagrams and notations. The Use case diagram is designed to see the working logic of the proposed system. The sequence diagram is designed to describe, how the client and the server interacts with each other when processing a content. The flow of the proposed system is described with the activity diagram. We know where the application starts and

when it ends after processing the keywords and the current URL link. This will help the programmers to implement the internal logic for the module in the given specification.

In this part of design phase, the design is carried out using the top-down strategy. First the major modules are identified. Then they are divided into sub modules so that each module at the lowest level would address a single function of the whole system. Each module design is explained detail.

This chapter tells us how the input module is design in getting the users requirements. The detailed input design provides as information regarding what are tools used in getting inputs and send to the server.

Output design is gives the user with good interacting option on the screen. The information delivered to the users through the information system. Useful output is essential to ensure the use and acceptance of the information system. Users often judge the merit of a system based upon its output. Productive output can only be achieved via close interaction with users. The output is designed in attractive and effective way that user can access them with a problem.

5.4 USE CASE DIAGRAM

A use case diagram is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The below diagram figure shows the overall use case diagram for credit card fraud detection.

A use case diagram is a type of behavioral diagram defined by the unified modeling language .its purpose is to present a graphical overview of the functionality povided by a system in terms of actors thier goals and any dependencies between those use cases.

The **USE CASE** diagram below describes the interaction between the customers and card issuers

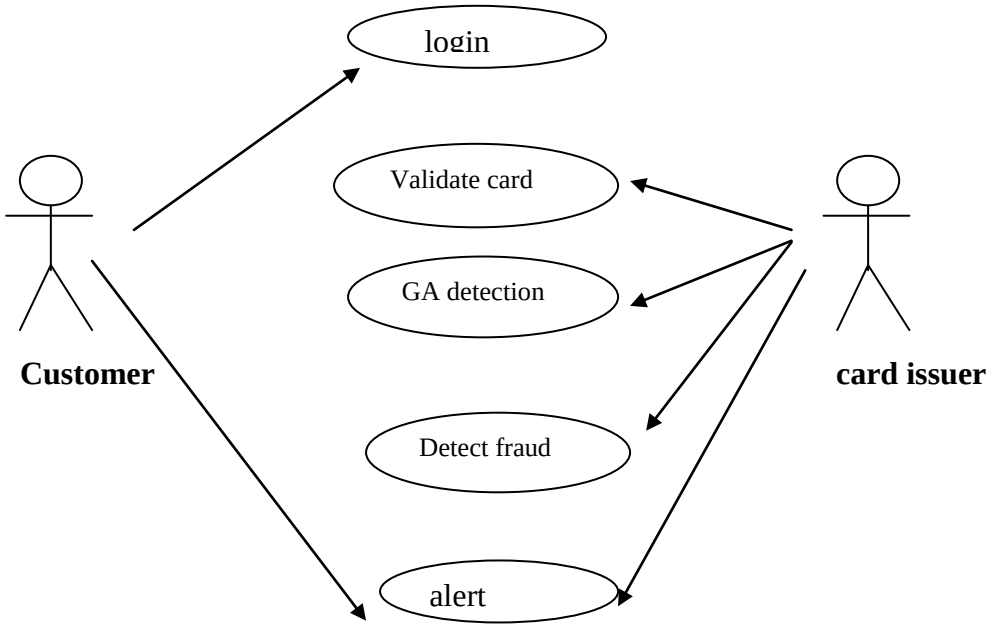


Figure 2 use case diagram of the overall structure

5.5 Flow of Genetic algorithm

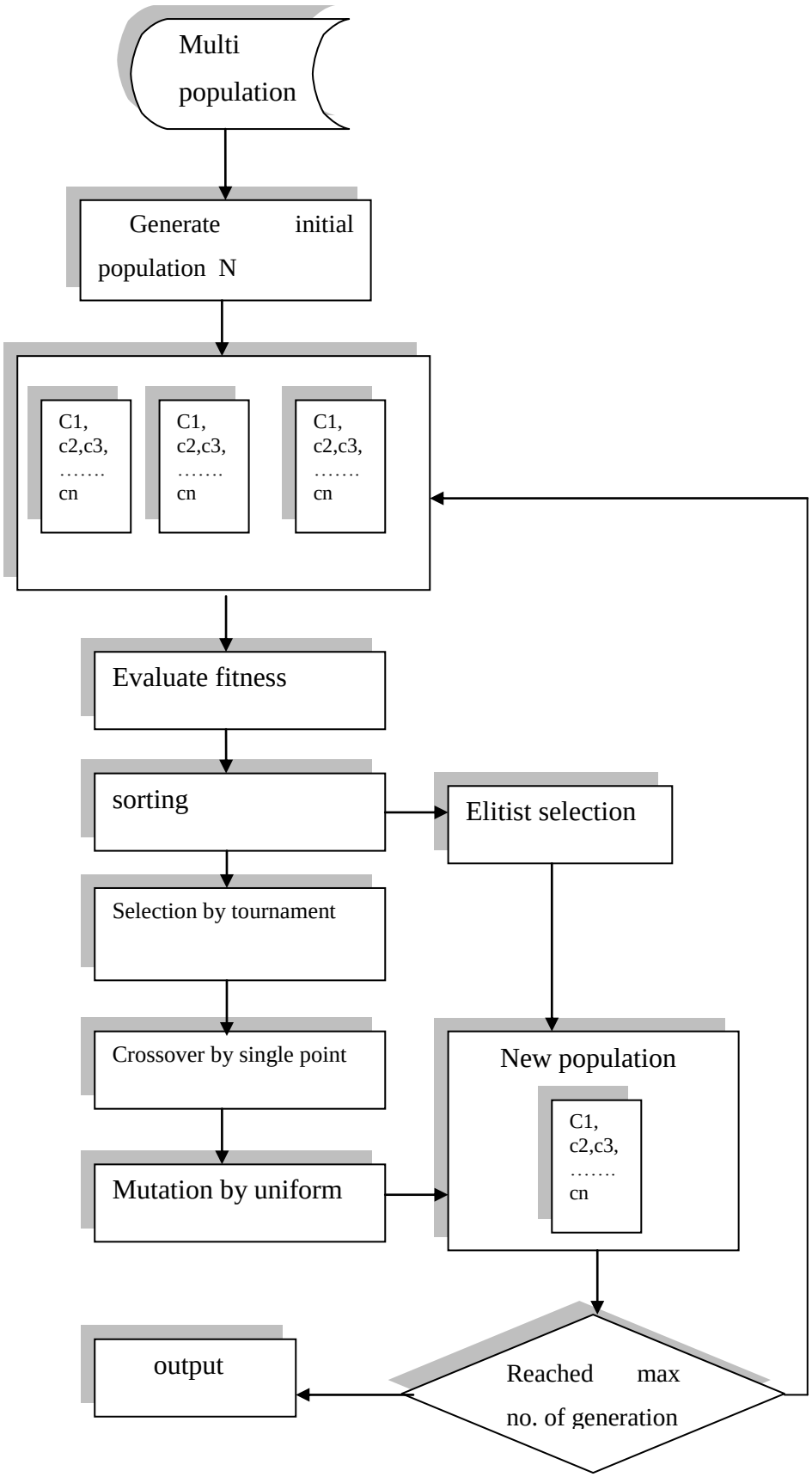


Figure 3 genetic algorithm

The above diagram states the process of genetic algorithm:

- Initially the initial population is selected randomly from the sample space which has many populations.
- The fitness value is calculated for each chromosome in each population and is sorted out.
- In selection process two parent chromosomes are selected through tournament method.
- The Crossover forms new offspring (children) from the parent chromosomes using single point probability.
- Mutation mutates the new offspring using uniform probability measure.
- In elitism selection the best solution are passed to the further generation.
- The new population is generated and undergoes the same process it maximum number of generation is reached.

CHAPTER 6

CODING, TESTING AND IMPLEMENTATION

It is the process of bringing developed system of revised system into operational use. If the implementation phase is not carefully planned and controlled, it can lead to many problems. Thus proper implementation is essential to provide a reliable system to meet managerial requirements.

6.1 IMPLEMENTATION

The application is completely written Java. This enables the credit card issuers to use this application across wide variety of devices independent of the vendor of the devices. We use oracle as a back end for storing database.

6.2 CODING

Standard coding practices are needed to ensure that the code is readable, understandable and easily modifiable. This project has defined standards and guidelines to be followed while pseudo coding. These standards were followed during the development of the application to produce code that is more consistent and to make code maintenance.

6.2.1 NAMING CONVENTIONS

Naming Conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier. All the controls used in the project where properly named according to their types.

6.2.2 COMMENTS

The comments are used in the programs to improve the understanding of the code in a clear way. The complete code was properly commented. Appropriate comments were given for each functions used, which described their functionality. Comments were also given for variable names to describe their purpose.

6.2.3 STATEMENTS CONSTRUCTION AND INDENTATION

Proper indentation has been done for nested blocks of code, function declaration, header files etc. The style of coding was carefully handled throughout the project as it encompasses a coding philosophy that stresses simplicity and clarity. Dernighan and Plauger states that, "Writing a computer program eventually boils down to writing a sequence of statements in the language at hand. How each of these statements is expressed determines in large measure the intelligibility of the whole..."

6.3 TESTING

Testing is one step in the software/web engineering process that could be viewed as destructives rather than constructive. Testing requires that the developer discard preconceived notions of the "correctness" of the software just developed and overcome a conflict of interest.

If testing is conducted successfully, it uncovers error in the software. As a secondary benefit testing demonstrates that software functions appear to be working according to specification, that performance requirements appear to have been met.

In addition data collected as testing is conducted provide a good indication of software reliability and some indication of software quality as whole. Testing cannot show the absence of defects, it can only show that software defects that are present.

The main objectives of testing are

- To ensure that during operation the system will perform as per specifications.
- To make sure that system meets the user requirements during operations.
- To make sure that during operation, incorrect input, processing and output will be detected.
- To see that when correct inputs are fed to the system the outputs are correct.
- Testing is a process of executing a program with the intent of finding errors.

6.3.1 UNIT TESTING

Developers write unit tests to check their own code. Unit testing differs from integration testing, which confirms that components work well together, and acceptance testing, which confirms that an application does what the customer expects it to do. Unit tests are so named because they test a single unit of code. Unit testing focuses verification effort on the smallest unit of software design. Each of the modules in this project was verified individually for errors.

6.3.2 INTEGRATION TESTING

Integration testing is a systematic testing for constructing the program structure while at the same time conducting tests to uncover errors associated within the interface. This testing was done with sample data. The need for integrated test is to find the overall system performances. The Integration testing can be performed in the credit card fraud detection as follows,In the Login Page User has not enter the Card Id and Pin number, but he/she clicks Sign In Button then the list of Errors should be displayed to the user as,

- Invalid Card ID
- Invalid Pin number

6.3.3 VALIDATION TESTING

Validation testing is where the requirements established as part of the software requirements analysis are validated against the software that has been constructed. It provides final assurance that the software meets all functional, behavioral and performance requirements. A deviation from the specification is uncovered and corrected. Each input field was tested with the validation rules specified integrity.

CODE

Credit card fraud.java

```
import java.applet.Applet;

import java.awt.Button;

import java.awt.Color;

import java.awt.FileDialog;

import java.awt.Font;

import java.awt.Frame;

import java.awt.Graphics;

import java.awt.Label;

import java.awt.TextArea;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.InputStreamReader;
```

```
import java.util.Arrays;


public class Creditcardfraud extends Applet implements ActionListener {
TextField brows,dname,dpath,key;
TextArea db,result,con;
Button browse,find,exit,clear;
int done;
Label browses,concl;
String us;
String strline = null;
String[] temp;
String[][] data = new String[50][50];
public void init(){
setBackground(Color.cyan);
setForeground(Color.magenta);
Label head=new Label("                                                    CREDIT
CARD FRAUD DETECTION SYSTEM
",Label.CENTER);
Font font = new Font("Serif", Font.ITALIC, 30);
head.setFont(font);
```

```
Label dataset=new Label("                    DATASET SELECTED
",Label.CENTER);


Label res=new Label("                                FRAUD DETECTED
",Label.CENTER);

browses = new Label("      Browse DataSet:     ", Label.LEFT);

concl = new Label("                                FRAUD TRANSACTIONS
", Label.LEFT);

brows = new TextField(50);

db = new TextArea(20,70);

result = new TextArea(20,70);

con = new TextArea(10,100);

browse = new Button("   Browse   ");

find = new Button("   Find     ");

exit = new Button("   Exit     ");

clear = new Button("   Clear     ");

brows.disable();

resize(1200,700);

Label l1 =new Label("

");
```

```
Label l2 =new Label("

                                ");

Label l3 =new Label("                                                                ");

Label l4 =new Label("
");

add(head);

setForeground(Color.BLUE);

add(l2);

add(browses);

add(brows);

setForeground(Color.BLACK);

add(browse);

add(find);

add(clear);

add(exit);

//          add(l1);

add(l4);

add(dataset);

add(l3);

add(res);
```

```
add(l3);

add(db);

add(result);

add(concl);

add(l3);

add(con);


//          register to receive action events

browse.addActionListener(this);

find.addActionListener(this);

exit.addActionListener(this);

}

public void actionPerformed(ActionEvent ae) {


String str = ae.getActionCommand();


if(str.equals("   Browse   ")) {

try

{

FileDialog fd = new FileDialog(new Frame(), "Please choose a file:", FileDialog.LOAD);
```

```
fd.show();

if (fd.getFile() != null) {

File f = new File(fd.getDirectory(), fd.getFile());

String path=f.getPath();

brows.setText(path);


FileInputStream fstream =new FileInputStream(path);

DataInputStream in =new DataInputStream(fstream);

BufferedReader br=new BufferedReader(new InputStreamReader(in));


int k=0;

for(int i=0;i<=20;i++)

{

strline=br.readLine();

temp =strline.split(",");


//        System.out.println(temp[0] +" "+temp[1] +" "+temp[2] +" "+temp[3] +" "+temp[4]);

//        System.out.println(" 1");


for(int j=0;j<=11;j++)
```

```
{
data[i]=temp;
}
}
for(int i=0;i<=20;i++)
{
for(int j=0;j<=11;j++)
{
db.append(data[i][j]);System.out.print(data[i][j]);
db.append("\t");System.out.print("\t");
}

if(i==0)
{
db.append("\n");
db.append("_______\t____\t_______\t__\t_______\t__\t_____\t___\t___\t____\t___\t____");
db.append("\n");
}
db.append("\n"); System.out.println(" ");
```

```
}



}

}

catch(Exception e)

{

System.out.println(e.toString());

}

}

else if(str.equals("   Exit     ")) {

System.exit(0);

System.out.println("\n cancel ");

//    repaint();

}

else if(str.equals("   Clear     ")) {

db.setText("    ");

result.setText(" ");

brows.setText(" ");

System.out.println("\n Clear ");

//    repaint();
```

```
}

else if(str.equals("   Find     ")) {

float[] res = null;

float[][] fre = new float[6][20];

float[][] loc = new float[6][20];

float[][] od = new float[6][20];

float[][] bb = new float[6][20];

float[][] ds = new float[6][20];

float[][] initPop = new float[21][5];

float[][] curPop = new float[21][5];

float[][] nexPop = new float[21][5];

float[][] finalPop = new float[21][5];

float[] resValue =new float[21];

Detection dt=new Detection();

Evaluate ev= new Evaluate();

NextGen ng= new NextGen();



/* CC usage Fequency */



int l=0,m=0;
```

```
result.append("Based on CC usage Fequency \n");

result.append("------------------------------ \n");

for(int i=1;i<=20;i++)

{

res= dt.ccfreq(data[i]);

if(res[0]>=1)

{

fre[l][m]=Float.valueOf(data[i][0]);m++;

fre[l][m]=res[1];

result.append("In CC ID: "+data[i][0]+" - Usage Freq. Fraud is found with value - "+res[1]);

result.append("\n");

l++;m=0;

}

initPop[i][0]=res[1];

}



/* CC usage Location */



l=0;m=0;

result.append("\n");
```

```
result.append("Based on CC usage Location \n");

result.append("------------------------------ \n");

for(int i=1;i<=20;i++)

{

res= dt.ccloc(data[i]);

if(res[0]>=1)

{

loc[l][m]=Float.valueOf(data[i][0]);m++;

loc[l][m]=res[1];

result.append("In CC ID: "+data[i][0]+" - Usage Location Fraud is found with value -
"+res[1]);

result.append("\n");

l++;m=0;

}

initPop[i][1]=res[1];

}



/* CC OverDraft */



l=0;m=0;

result.append("\n");
```

```
result.append("Based on CC OverDraft \n");

result.append("------------------------------ \n");

for(int i=1;i<=20;i++)

{

res= dt.ccod(data[i]);

if(res[0]>=1)

{

od[l][m]=Float.valueOf(data[i][0]);m++;

od[l][m]=res[1];

result.append("In CC ID: "+data[i][0]+" - CC OverDraft Fraud is found with value -
"+res[1]);

result.append("\n");

l++;m=0;

}

initPop[i][2]=res[1];

}



/* Current Book Balance */



l=0;m=0;

result.append("\n");
```

```
result.append("Based on CC Book Balance \n");

result.append("------------------------------ \n");

for(int i=1;i<=20;i++)

{

res= dt.ccbb(data[i]);

if(res[0]>=1)

{

bb[l][m]=Float.valueOf(data[i][0]);m++;

bb[l][m]=res[1];

result.append("In CC ID: "+data[i][0]+" - CC Book Balance Fraud is found with value -
"+res[1]);

result.append("\n");

l++;m=0;

}

initPop[i][3]=res[1];

}



/* Average Daily Spending */



l=0;m=0;

result.append("\n");
```

```
result.append("Based on CC Average Daily Spending \n");

result.append("---------------------------------- \n");

for(int i=1;i<=20;i++)

{

res= dt.ccds(data[i]);

if(res[0]>=1)

{

ds[l][m]=Float.valueOf(data[i][0]);m++;

ds[l][m]=res[1];

result.append("In CC ID: "+data[i][0]+" - CC Daily Spending Fraud is found with value -
"+res[1]);

result.append("\n");

l++;m=0;

}

initPop[i][4]=res[1];

}



//          float[][] finalresult = dt.organize(fre,loc,od,bb,ds);



for(int i=1;i<=20;i++)

{
```

```
for(int j=0;j<=4;j++)

{

System.out.print(initPop[i][j]);

System.out.print("\t ");

}

System.out.println("");                                    }

System.out.println("********** end of INIT Population ");



curPop=initPop;



for(int q=1;q<=20;q++)

{

nexPop=ng.getNextGen(curPop);

System.out.println(" \n");

System.out.println(" Current Popoulation - Generation -  "+q);

System.out.println("_____________________________________________ \n");

for(int i=1;i<=20;i++)

{

for(int j=0;j<=4;j++)

{
```

```
System.out.print(nexPop[i][j]);

System.out.print("\t ");

}

System.out.println(" ");

}

curPop=nexPop;



System.out.println(" \n\n Critical Values Found after Limited number of Generations (sorted order)");



resValue = dt.resValue(curPop);

Arrays.sort(resValue);

for(int i=1;i<=20;i++)

{

System.out.println(resValue[i]);

}

}

float criti=resValue[15];

float monit=resValue[10];

float ordin=resValue[5];
```

```
System.out.println("\n\n Critical Values of each transaction of given DataSet");

System.out.println(" ---------------------------------------------------------- ");

float[][] finalresult = dt.organize(fre,loc,od,bb,ds);



System.out.println("\n\n Value of Critic, Monitor and Ordinary Faruds");

System.out.println("\n\n "+criti+"  "+monit+"   "+ordin);

System.out.println(" \n\n Fraud Detected used Genetic Algorithm: ");

System.out.println("------------------------------------------- ");

con.append(" Critical Fraud Detected: ");con.append("\n");

con.append("------------------------------------ ");

System.out.println("Critical Fraud Detected:  ");

System.out.println("------------------------------------ ");

for(int i=0;i<=19;i++)

{

if((finalresult[i][2])>= criti)

{

con.append("\n");

con.append(" Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);

System.out.println(" Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);
```

```
System.out.println(" ");

con.append("\n");

}

}

con.append(" \n Monitorable Fraud Detected: ");con.append("\n");

con.append("---------------------------------------- ");

System.out.println("Monitorable Fraud Detected:  ");

System.out.println("------------------------------- ");


for(int i=0;i<=19;i++)

{

if(((finalresult[i][2])>= monit) && ((finalresult[i][2])< criti))

{

con.append("\n");

con.append(" Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);

System.out.println("Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);

System.out.println(" ");

con.append("\n");

}
```

```
}


con.append(" \n Ordinary Fraud Detected: ");con.append("\n");

con.append("------------------------------------- ");

System.out.println("Ordinary Fraud Detected:  ");

System.out.println("------------------------------------- ");

for(int i=0;i<=19;i++)

{

if(((finalresult[i][2])>= ordin) && ((finalresult[i][2])< monit))

{

con.append("\n");

con.append(" Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);

System.out.println("Credit Card with ID "+finalresult[i][0]+" is detected as fraud with
"+finalresult[i][1]+" occurance and its crical value is "+ finalresult[i][2]);

con.append("\n");

}

}

repaint();

}

}
```

```
public void paint(Graphics g) {

System.out.println(done);

if(done==1)

{

setForeground(Color.BLUE);

g.drawString("SUCCESS", 10, 190);

String msg="The File is Encrypted Successfully";

g.drawString(msg, 20, 205);

}

if(done==2)

{

setForeground(Color.RED);

g.drawString("ERROR", 10, 190);



String msg="The File is not Encrypted Successfully \n";

g.drawString(msg, 20, 205);

/*     if(!errmsg1.equals(null))

{

g.drawString(errmsg1, 20, 220);
```

```
}
if(!errmsg2.equals(null))
{
g.drawString(errmsg2,20, 235);
}*/
System.out.println("\n paint ");
}


}
}
```

Detection.java

```
public class Detection {
public float[] ccfreq(String[] a) {
float[] res = new float[2];
String[] temp =a;
res[0]=0; res[1]=0;
//        for(int j=0;j<=11;j++)
//        System.out.println(temp[j]);
```

```
float ccfreq =Float.valueOf(temp[3])/Float.valueOf(temp[6]);
if(ccfreq>0.2)
{
if(Float.valueOf(temp[7])>(5*ccfreq))
{
res[0]=1;
res[1]=(Float.valueOf(temp[7])*ccfreq);
}
}
//          System.out.println("  "+ccfreq+"   "+res[0]+"   "+res[1]);
if(res[0]<1)
{
res[1]=(float)ccfreq;
}
return res;
}

public float[] ccloc(String[] a) {
float[] res = new float[2];
String[] temp =a;
```

```
res[0]=0; res[1]=0;
int loc=Integer.valueOf(temp[8]);
if((loc<= 5) && (Integer.valueOf(temp[9])>( 2 * loc)))
{
res[0]=1;
res[1]=(Float.valueOf(loc)/ Float.valueOf(temp[9]));
}
if(res[0]<1)
{
res[1]=(float)0.01;
}

return res;
}


public float[] ccod(String[] a) {
float[] res = new float[2];
String[] temp =a;
res[0]=0; res[1]=0;
```

```
float od =Float.valueOf(temp[5])/Float.valueOf(temp[3]);
if(od<=0.2)
{
if(Float.valueOf(temp[10])>=1)
{
res[0]=1;
res[1]=(Float.valueOf(temp[10])*od);
}
}
if(res[0]<1)
{
res[1]=(float)od;
}
//          System.out.println("  "+od+"   "+res[0]+"   "+res[1]);
return res;
}



public float[] ccbb(String[] a) {
float[] res = new float[2];
```

```
String[] temp =a;
res[0]=0; res[1]=0;
float bb =Float.valueOf(temp[2])/Float.valueOf(temp[4]);
if(bb<=0.25)
{
res[0]=1;
res[1]=(Float.valueOf(2)*bb);
}
if(res[0]<1)
{
res[1]=(float)bb;
}
//          System.out.println("  "+bb+"   "+res[0]+"   "+res[1]);
return res;
}



public float[] ccds(String[] a) {
float[] res = new float[2];
String[] temp =a;
```

```
res[0]=0; res[1]=0;
float mon= Float.valueOf(temp[6])/30;
float bal= 100000 - Float.valueOf(temp[4]);
float tot = mon*bal;
float ds =tot/Float.valueOf(temp[6]);
if((10*ds)<Float.valueOf(temp[11]))
{
res[0]=1;
if(Float.valueOf(temp[11])>0)
res[1]=(Float.valueOf(temp[11])/ (10*ds));
else
res[1]=(float) 0.0;


}
if(res[0]<1)
{
res[1]=(float)0.01;
}
//          System.out.println("  "+ds+"   "+res[0]+"   "+res[1]);
return res;
```

```
}



public float[][] organize(float[][] fre,float[][] loc,float[][] od,float[][] bb,float[][] ds)

{

//        System.out.println(" ");

//        System.out.println(" final");

float[][] result=new float[20][20];

float now;

float id=Float.valueOf(11111),val=(float)0;

int ins=0,z=0;



for(int i=0;i<=19;i++)

{

now =id;

//        System.out.println(now);

for(int j=0;j<=4;j++)

{

if(fre[j][0]==now){

ins++;
```

```
val=val+fre[j][1];
}

//        System.out.println(ins);
//        System.out.println(val);

if(loc[j][0]==now){
ins++;
val=val+loc[j][1];
}

if(od[j][0]==now){
ins++;
val=val+od[j][1];
}

if(bb[j][0]==now){
ins++;
val=val+bb[j][1];
}
```

```
if(ds[j][0]==now){
ins++;
val=val+ds[j][1];
}

}

result[z][0]=now;
result[z][1]=ins;
result[z][2]=val;
ins=0;val=0;z++;id++;
}

System.out.println(" AccountNo\tFraud Occurance\tCritical Value");
for(int i=0;i<=19;i++)
{
for(int j=0;j<=2;j++)
{
System.out.print(result[i][j]);
```

```
System.out.print("\t    ");

}

System.out.println(" ");

}


//          System.out.println("********** end of organise Result ************ ");


return result;

}


public float[] resValue(float[][] a)

{

float[] res=new float[21];

float[][] b=new float[21][6];

b=a;

float sum =0;


for(int i=1;i<=20;i++)

{

for(int j=0;j<=4;j++)
```

```
{
sum=sum+b[i][j];
}
res[i]=sum;
sum=0;
}

return res;
}
}
```

Evaluate.java

```
public class Evaluate {
public int[] findElite(float[][] a) {
int[] res = new int[2];
float[][] temp =new float[21][5];
temp=a;
float sum=0,sum1=0,sum2=0;
int e1=0,e2=0;
```

```
for(int i=1;i<=20;i++)
{
sum=temp[i][0]+temp[i][1]+temp[i][2]+temp[i][3]+temp[i][4];

if(i==1)
{
sum1=sum;
e1=1;
}
if(i==2)
{
sum2=sum;
e2=2;

if(sum1<sum2)
{
sum1=sum1+sum2;
sum2=sum1-sum2;
sum1=sum1-sum2;
e1=2;
```

```
e2=1;
}
}

if((sum>sum1) && (sum>sum2))
{
sum2=sum1;
sum1=sum;

e2=e1;
e1=i;

}
else if(sum>sum2)
{
sum2=sum;
e2=i;
}

}
```

```
System.out.println("\n");

res[0]=e1;

res[1]=e2;

System.out.println(" Elitist Value from the Previous Population \n");

System.out.println("  "+e1+"   "+sum1+"   "+e2+"   "+sum2);

return res;

}




}
```

Nextgen.java

```
import java.util.Random;


public class NextGen {

static int u=0;



public float[][] getNextGen(float[][] a)

{
```

```
float[][] curPop=new float[21][6];

curPop=a;

float[][] temp1 = new float [21][5];

float [][] res=new float[21][5];

float[][] temp =new float[21][5];

float[] x=new float[5];

float[] y=new float[5];

float[] z=new float[5];

int cur=1;

Evaluate ev = new Evaluate();

int[] elite = new int[2];

if(u<1)

{

System.out.println(" ***************");

elite = ev.findElite(curPop);

u=1;

}

else

{

System.out.println(" *************** ");
```

```
elite = ev.findElite(curPop);

}

int b=elite[0], c=elite[1];

res[cur]=curPop[b];cur++;

res[cur]=curPop[c];cur++;

//cur=2;

Random randomGen = new Random();

Operators op=new Operators();


for(int i=0;i<=8;i++)

{


int rand = randomGen.nextInt(19);

int rand1 = randomGen.nextInt(19);


if(rand==0)

rand=rand1;

if(rand1==0)

rand1=rand;
```

```
if(rand==rand1)

if(rand>1)rand1=rand-1;

else rand1=rand+1;



//          System.out.println(" \n\n\n");

//          System.out.println("the random rows are "+rand+"    "+rand1);

temp1=curPop;

//          System.out.println(" ******start of crossover*** ");

z = op.crossover(temp1,rand,rand1);

//          System.out.println("z values"+z[0]+"  "+z[1]+"  "+z[2]+"  "+z[3]+"  "+z[4]);

float[] m =new float[5]; //=z;

m[0]=z[0];m[1]=z[1];m[2]=z[2];m[3]=z[3];m[4]=z[4];

x=z;

y = op.mutat(x);

//          System.out.println("y values"+y[0]+"  "+y[1]+"  "+y[2]+"  "+y[3]+"  "+y[4]);

float[] n =new float[5];

n[0]=y[0];n[1]=y[1];n[2]=y[2];n[3]=y[3];n[4]=y[4];



//                    System.out.println("last ");
```

```
//        System.out.println(m[0]+" "+m[1]+" "+m[2]+" "+m[3]+" "+m[4]);

//        System.out.println(n[0]+" "+n[1]+" "+n[2]+" "+n[3]+" "+n[4]);

//        System.out.println(z[0]+" "+z[1]+" "+z[2]+" "+z[3]+" "+z[4]);

//        System.out.println(y[0]+" "+y[1]+" "+y[2]+" "+y[3]+" "+y[4]);


res[cur]=m;cur++;

res[cur]=n;cur++;

}

return res;

}

}
```

Operators.java

```
import java.util.Random;


public class Operators {

public float[] mutat(float[] a)

{

float[] res1 = null, parent1,child;
```

```
parent1=a;


Random randomGen = new Random();

int rand = randomGen.nextInt(4);

int rand1 = randomGen.nextInt(4);


if(rand==rand1)

if(rand>1)rand1=rand-1;

else rand1=rand+1;


//          System.out.println(" "+ parent1[0]+" "+ parent1[1]+" "+ parent1[2]+" "+
parent1[3]+" "+ parent1[4]);

//    System.out.println("********** end of Parent 1");


//   System.out.println("the random num are "+rand+"  "+rand1);


child=parent1;


if((rand==0) || (rand1==0))

{
```

```
float t1 = randomGen.nextFloat();

int t2 = randomGen.nextInt(6);

if(t1>0.2)t1=(float)0.15;

if(t2==0) t2=6;


child[0]=t1*t2;

}


if((rand==1) || (rand1==1))

{

int t1 = randomGen.nextInt(4);

int t2 = randomGen.nextInt(14);

if(t1==0) t1=4;

if(t2==0) t2=4;

child[1]=(float)((float)t1/(float)t2);

}

if((rand==2) || (rand1==2))

{

float t1 = randomGen.nextFloat();

int t2 = randomGen.nextInt(1);
```

```
if(t1>0.2)t1=(float)0.16;

if(t2==0) t2=(int) 1;


child[2]=t1*t2;




}

if((rand==3) || (rand1==3))

{

float t1 = randomGen.nextFloat();

if(t1>0.2)t1=(float)0.2;


child[3]=t1*2;


}

if((rand==4) || (rand1==4))

{


float t1 = randomGen.nextFloat();

int t2 = randomGen.nextInt(1);
```

```
if(t1>0.2)t1=(float)0.15;

if(t2==0) t2=(int) 1;



child[4]=t1*t2;

}



//  child=parent1;



res1=child;



//   System.out.println("child of mutat "+ child[0]+" "+ child[1]+" "+ child[2]+" "+ child[3]+"
"+ child[4]);



//   System.out.println("******child 2*** ");



return res1;

}

public float[] crossover(float[][] a,int b,int c)

{

float[] res = null, parent1, parent2,child;
```

```
parent1=a[b];

parent2=a[c];


Random randomGen = new Random();

int rand = randomGen.nextInt(4);

if(rand==4) rand=3;


child=parent1;


if(rand==0)

{

child[0]=parent1[0];child[1]=parent2[1];child[2]=parent2[2];child[3]=parent2[3];child[4]=pa
rent2[4];

}


if(rand==1)

{

child[0]=parent1[0];child[1]=parent1[1];child[2]=parent2[2];child[3]=parent2[3];child[4]=pa
rent2[4];

}

if(rand==2)
```

```
{

child[0]=parent1[0];child[1]=parent1[1];child[2]=parent1[2];child[3]=parent2[3];child[4]=pa
rent2[4];

}

if(rand==3)

{

child[0]=parent1[0];child[1]=parent1[1];child[2]=parent1[2];child[3]=parent1[3];child[4]=pa
rent2[4];

}

//          System.out.println("child of cross "+ child[0]+" "+ child[1]+" "+ child[2]+" "+
child[3]+" "+ child[4]);

//          System.out.println("******child 1*** ");

res=child;



return res;

}
```

CHAPTER 7

CONCLUSION AND FORESEEABLE ENHANCEMENTS

7.1 CONCLUSION

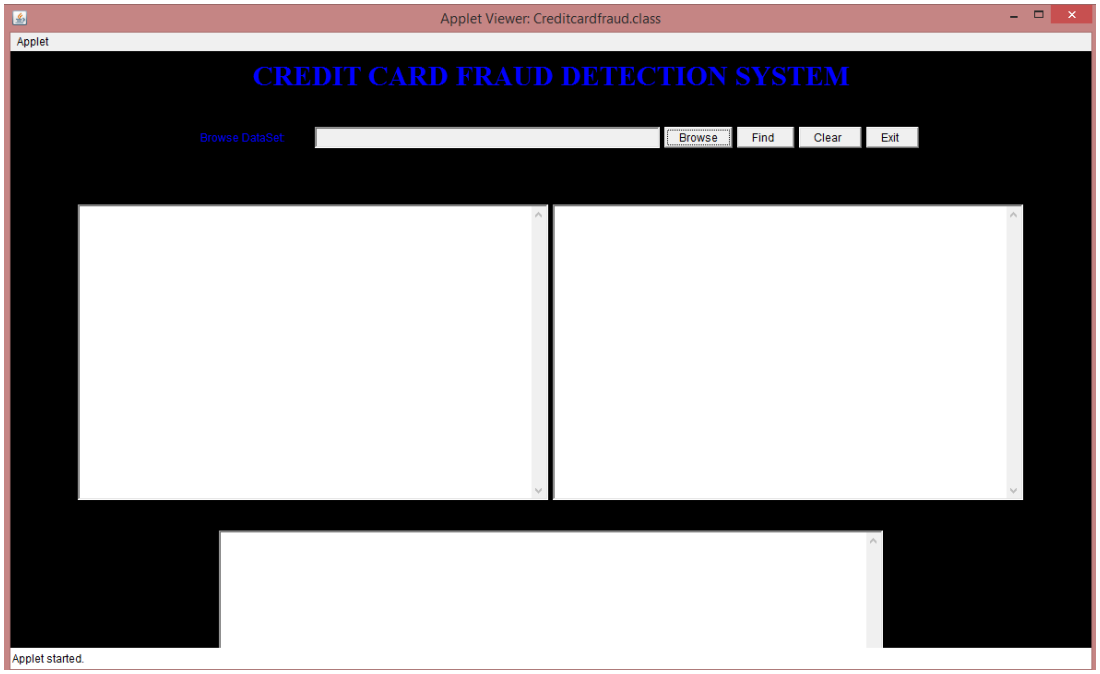
This method proves accurate in deducting fraudulent transaction and minimizing the number of false alert. Genetic algorithm is a novel one in this literature in terms of application domain. If this algorithm is applied into bank credit card fraud detection system, the probability of fraud transactions can be predicted soon after credit card transactions. And a series of anti-fraud strategies can be adopted to prevent banks from great losses and reduce risks.

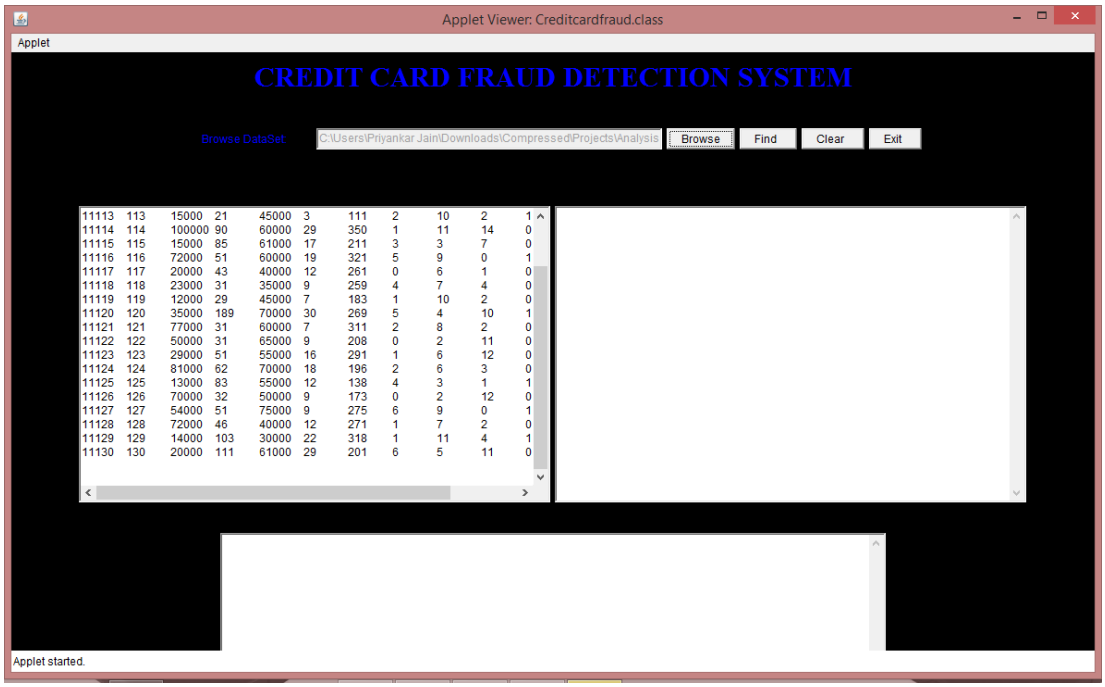
The objective of the study was taken differently than the typical classification problems in that we had a variable misclassification cost. As the standard data mining algorithms does not fit well with this situation we decided to use multi population genetic algorithm to obtain an optimized parameter.

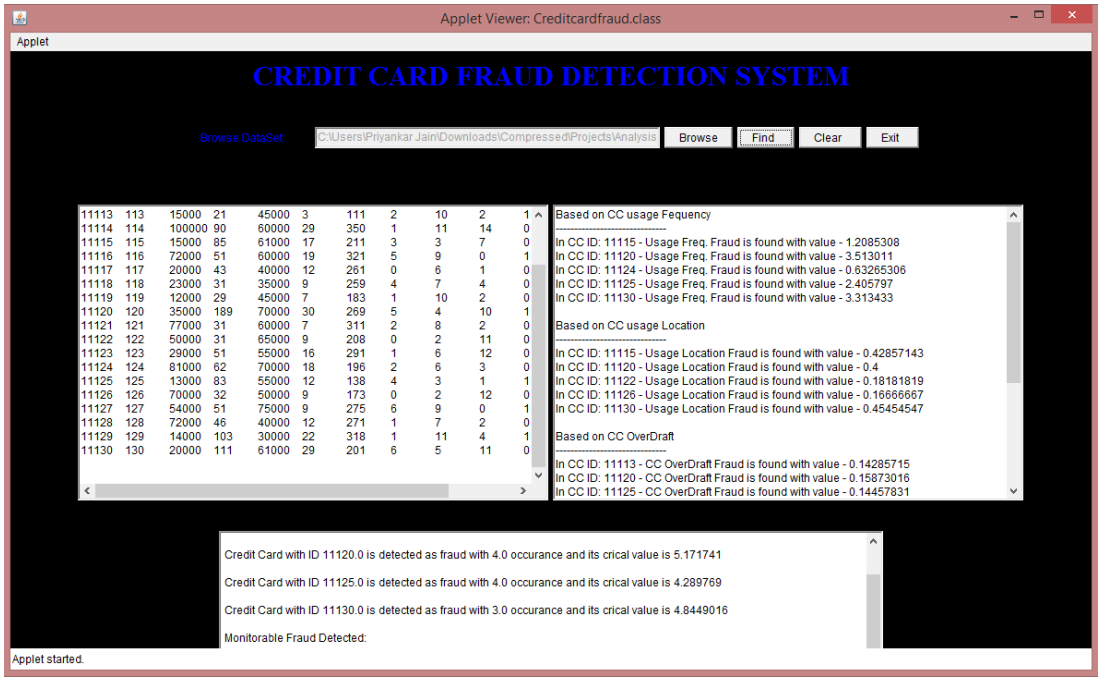
7.2 FUTURE ENHANCEMENTS

The findings obtained here may not be generalized to the global fraud detection problem. As future work, some effective algorithm which can perform well for the classification problem with variable misclassification costs could be developed.

SCREEN SHOTS






Applet Viewer: Creditcardfraud.class
Applet
CREDIT CARD FRAUD DETECTION SYSTEM
Browse DataSet
C:\Users\Priyankar Jain\Downloads\Compressed\Projects\Analysis
Browse
Find
Clear
Exit
11113 113 15000 21 45000 3 111 2 10 2 1
11114 114 100000 90 60000 29 350 1 11 14 0
11115 115 15000 85 61000 17 211 3 3 7 0
11116 116 72000 51 60000 19 321 5 9 0 1
11117 117 20000 43 40000 12 261 0 6 1 0
11118 118 23000 31 35000 9 259 4 7 4 0
11119 119 12000 29 45000 7 183 1 10 2 0
11120 120 35000 189 70000 30 269 5 4 10 1
11121 121 77000 31 60000 7 311 2 8 2 0
11122 122 50000 31 65000 9 208 0 2 11 0
11123 123 29000 51 55000 16 291 1 6 12 0
11124 124 81000 62 70000 18 196 2 6 3 0
11125 125 13000 83 55000 12 138 4 3 1 1
11126 126 70000 32 50000 9 173 0 2 12 0
11127 127 54000 51 75000 9 275 6 9 0 1
11128 128 72000 46 40000 12 271 1 7 2 0
11129 129 14000 103 30000 22 318 1 11 4 1
11130 130 20000 111 61000 29 201 6 5 11 0
Based on CC usage Fequency
In CC ID: 11115 - Usage Freq. Fraud is found with value - 1.2085308
In CC ID: 11120 - Usage Freq. Fraud is found with value - 3.513011
In CC ID: 11124 - Usage Freq. Fraud is found with value - 0.63265306
In CC ID: 11125 - Usage Freq. Fraud is found with value - 2.405797
In CC ID: 11130 - Usage Freq. Fraud is found with value - 3.313433
Based on CC usage Location
In CC ID: 11115 - Usage Location Fraud is found with value - 0.42857143
In CC ID: 11120 - Usage Location Fraud is found with value - 0.4
In CC ID: 11122 - Usage Location Fraud is found with value - 0.18181819
In CC ID: 11126 - Usage Location Fraud is found with value - 0.16666667
In CC ID: 11130 - Usage Location Fraud is found with value - 0.45454547
Based on CC OverDraft
In CC ID: 11113 - CC OverDraft Fraud is found with value - 0.14285715
In CC ID: 11120 - CC OverDraft Fraud is found with value - 0.15873016
In CC ID: 11125 - CC OverDraft Fraud is found with value - 0.14457831
Credit Card with ID 11120.0 is detected as fraud with 4.0 occurance and its crical value is 5.171741
Credit Card with ID 11125.0 is detected as fraud with 4.0 occurance and its crical value is 4.289769
Credit Card with ID 11130.0 is detected as fraud with 3.0 occurance and its crical value is 4.8449016
Monitorable Fraud Detected:
Applet started.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES

Papers:

[1] M. Hamdi Ozcelik, Ekrem Duman, Mine Isik, Tugba Cevik, Improving a credit card fraud detection system using genetic algorithm, International conference on Networking and information technology 2010.

[2] Wen-Fang YU, Na Wang, Research on Credit Card Fraud Detection Model Based on Distance Sum, IEEE International Joint Conference on Artificial Intelligence 2009.

[3] clifton phua, vincent lee1, kate smith & ross gayler, A Comprehensive Survey of Data Mining-based Fraud Detection Research,2005.

[4] Elio Lozano, Edgar Acu˜na, Parallel algorithms for distance-based and density-based outliers,2006.

[5] Credit card fraud detection using hidden markov model – Abinav Srivastava,Amlan Kundu,Shamik Sural,Arun K.majumdar

Websites:

[1]http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html

[2] http://www.kxcad.net/cae_MATLAB/toolbox/gads/f6691.html

[3]http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/front.html

[4]http://www.easywayserver.com/blog/user-login-in-jsp/

[5]http://www.faqs.org/patents/app/20100094765

Textbooks:

[1]Pressman, Roger S. Software engineering: a practitioner's approach / Roger S. Pressman.—5th ed.p. cm (McGraw-Hill series in computer science).

[2]E.Balagurasamy, Programming with java, Tata McGraw-Hill Publication.

[3]Ali Bahrami,Oject Oriented system Development, Tata McGraw-Hill Publication .

[4]Jiwei Han et,al., "Data Mining :Concepts and Techniques", MorganKaufmaan Series,2000