1. INTRODUCTION

1.1 OVERVIEW

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 PURPOSE

credit card fraud detection system is to identify suspicious events and report them to an analyst while letting normal transactions be automatically processed. For years, financial institutions have been entrusting this task to rule-based systems that employ rule sets written by experts.

2. LITERATURE SURVEY

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.

2.1 EXISTING SOLUTION

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 discriminate 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.

2.2 PROPOSED SOLUTION 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.

3. THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM

3.2 HARDWARE/SOFTWARE REQUIREMENTS: 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. SOFTWARE REQUIREMENTS • Application software Framework: Java • Back End: SQL Server • Operating System: Windows XP Professional or more 4. EXPERIMENTAL INVESTIGATIONS In this experimental study, we discovered several shortcomings of existing methods. We found that the approaches designed specifically to tackle the imbalance problem are not adequately effective. While these approaches improve sensitivity, this improvement leads to an increase in the number of false alarm and thus also to a drop in accuracy and AUPRC. Practically speaking, this can be costly for the financial institution just the same way a fraud event costs. Even a minimal deterioration of accuracy, say 1% hides a large misclassification rate of the majority group. The problem is summarized as follows: using imbalanced classification approaches, the number of false alarms generated is higher than the number of frauds that are detected. The results of this experimental study greatly motivated us to explore the other methods that focus on detecting the hidden patterns of fraud, with minimum misclassification.5. FLOWCHART 6. RESULT The reason behind using a small dataset is that the computation cost for large datasets is high; a high-performance system is required to conduct experiments on larger dataset. The system we used for our experiments was rather a simple system and hence not able to handle high computation costs. Besides, multiple parameters affect the accuracy of this model, such as the threshold and the size of the detector set. We found a set of 2000 detectors; yet a bigger set is needed to achieve better results that would increase training time and computation cost significantly7. ADVANTAGES AND DISADVANTAGES • Non- availability of real data set • Unbalanced data set • Size of data set • Determining the appropriate evaluation parameters 8. APPLICATIONS • Applications for Issuing Banks • Applications

for Merchants and Retailers • Establishing these relationships between identities also helps build a customer's reputation score and is the basis to approve or disapproved transactions, according to the company. • A mong the online activities and other details the platform checks are user purchases, transactions and orders; email, billing, shipping and IP address; payment methods; custom data and technology devices used. It also gathers information and analyzes highly-detailed behavioral patterns such as browsing patterns, keyboard preferences and screen tilt. 9. CONCLUSION This method proves accurate in deducting fraudulent transaction and minimizing the number of false alerts. 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 parameter10.FUTURE SCOPE 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. 11.BIBILOGRAPHY 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 Miningbased Fraud Detection Research,2005. [4] Elio Lozano, Edgar Acuna, 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 APPENDIX SOURCE 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]; [3:03 PM, 5/8/2021] ganesh p cvr clg: 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]); [3:04 PM, 5/8/2021] ganesh p cvr clg: 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 "); } }
```