

CREDIT CARD FRAUD DETECTION USING NEURAL NETWORK

PROJECT REPORT

by

**VIGNESH G
18BEC0692**

**R SRIVATSAN
18BEC0461**

**RISHABH KARTHIK RAMESH
18BEC0662**

**SHIVAJI CHOWDARY MOPARTHY
18BEC0707**

School of Electronics
Vellore Institute of Technology Vellore



June 2021

Table of Contents:

Declaration by authors:	3
Abstract:	4
1. Introduction:	5
1.1 Motivation	5
1.2 Aim	5
1.3 Objective	5
2. Literature Survey:	6
3. Proposed System Requirement Analysis and Design:	7
3.1 Introduction	7
3.2 Requirement Analysis:	7
3.2.1 Stakeholder Identification	7
3.2.2 Functional Requirements	7
3.2.3 Non-Functional Requirements	8
3.2.4 System Requirements.....	8
3.2.4.1 Hardware Requirements.....	8
3.2.4.2 Software Requirements	8
3.2.5 Software Requirement Specification Document	9
3.2.6 Work Breakdown Structure	10
4 Design of Proposed System:	12
4.1 Introduction	12
4.2 High Level Design	12
4.2.1 Architecture Diagram.....	12
4.2.2 Architecture Diagram Explanation	13
4.2.3 UI Design	13
4.3.1 ER Diagram	13
4.3.2 UML Diagram	15
5. Implementation and Testing:	18
6. Conclusion, Limitaitons and Future Work:	22
7. References:	22

Declaration by Authors

This is to declare that this report has been written by us as part of our coursework. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. We aver that if any part of the report is found to be plagiarized, we shall take full responsibility for it.

VIGNESH G

18BEC0692

R SRIVATSAN

18BEC0461

SHIVAJI CHOWDARY MOPARTHY

18BEC0707

RISHABH KARTHIK RAMESH

18BEC0662

Date: June 2021

ABSTRACT

The credit card business has increased speedily over the last two decades. Corporations and establishments are moving towards various online services, which aims to permit their customers with high potency and accessibility. The evolution is a huge step towards potency, accessibility, and profitableness of view. Nevertheless, it additionally has some downsides. These smart services are recently prone to significant security related vulnerabilities. Developing business through card depends on the fact that neither the card nor the user needs to be present at the point of transaction. Thus, it is impossible for merchandiser to check whether the cardholder is real or not. Companies' loss in recent times are majorly due to the credit card fraud and the fraudsters who ceaselessly obtain new ways to commit the unlawful activities. As we know that Artificial Neural Network (ANN) has the ability to work as a human brain when trained properly. So, we will be implementing a neural network along with Self Organizing Map (SOM) and discuss about its performance and accuracy.

Credit Card extortion are defined as the frauds which take place using the Credit Cards or Debit Cards particularly in the online transaction. Fraud detection methods are designed with an objective to prevent the online frauds in the system. By analysing existing data and detecting fraud is the foremost among the best techniques to reduce the successful frauds. Any variation in "usual spending" will also help to determine frauds. Some ways in which the fraudster can commit frauds are Merchant Related Fraud, Site Cloning, False Merchant Site, Credit Card Generators etc.

Keywords: Artificial Neural Network (ANN), Machine Learning, Self-Organising Maps (SOM).

INTRODUCTION

1.1 Motivation

In many places credit card frauds and banking frauds have increased significantly causing a lot of problems for the users and also banks to keep their money safe and also detecting these frauds have been very difficult as the pattern of frauds keep on changing hence, we have thought of developing an application that can detect frauds automatically irrespective of the trends followed.

1.2 Aim

The aim of the project is to create a simple web application tool for credit card fraud detection that will help users to keep their credit card safe and also for banks to maintain a secure application and keep their money safe from frauds.

1.3 Objective

Fraud detection methods are designed with an objective to prevent the online frauds in the system. By analysing existing data and detecting fraud is the foremost among the best techniques to reduce the successful frauds. Any variation in “usual spending” will also help to determine frauds. Since human tend to follow specific behaviouristic profile such as the category of shopping. SOMs are well known applied method for classification on transactions.

The main modules of our project is

1.Detecting of frauds

2.Prevention of fraud by generating probability report

3.Customer can automatically enter details and check if their credit card is under protection and also no unusual transactions occur.

LITERATURE SURVEY

1)Credit card fraud detection based on whale algorithm optimized BP neural network –The 13th International Conference on Computer Science & Education (ICCSE 2018) August 8-11, 2018. Colombo, Sri Lanka-the advantages of BP neural network algorithm and whale algorithm, and proposes a new credit card fraud detection algorithm using whale algorithm to optimize BP neural network algorithm. Matlab simulation results show that the WOA-BP algorithm proposed in this paper has high detection accuracy and fast convergence speed, which improves the accuracy of credit card fraud detection.

2) Credit Card Fraud Detection using Artificial Neural Network and BackPropagation.Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2020) IEEE Xplore Part Number: CFP20K74-ART; ISBN: 978-1-7281-4876-2. The BackPropagation is used in this which is still in research in the area of Artificial Intelligence. This model can detect transactions in real-time. So, banks can detect frauds and stop the ongoing transactions if that transaction is a fraud. Then the customer doesn't have to wait to know if someone has misused a card or not.

3) A Deep Neural Network Algorithm for Detecting Credit Card Fraud.2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). This paper presents a deep neural network algorithm to detect credit card fraud.

4) Review On Fraud Detection Methods in Credit Card Transactions.2017 International Conference on Intelligent Computing and Control (I2C2'17). To detect fraud behaviour, bank and credit card companies are using various methods of data mining such as decision tree, rule based mining, neural network, fuzzy clustering approach, hidden Markov model or hybrid approach of these methods. This paper studies the different methods.

5) A.J. Graaff A.P. Engelbrecht “The Artificial Immune System for Fraud Detection in the Telecommunications Environment”; (2014)-KNN algorithm and HIDDEN MARKOV MODEL is implemented to optimize the best solution for the problem. This approach is proved to minimize the false alarm rates and increase the fraud detection rate. Moreover, the behaviour analysis process of the HMM method helps in minimizing the fraud rates thus retaliate further fraudulent activities more efficiently.

6) “Markova Scheme for Credit Card Fraud Detection”. International Conference on Advanced Computing, Communication and Networks; (2011). An HMM is initially trained with the normal behaviour of a cardholder. If an Incoming credit card transaction is not accepted by the trained HMM with sufficiently high probability, it is considered to be fraudulent. At the same time, we try to ensure that genuine transactions are not rejected.

PROPOSED SYSTEM REQUIREMENT ANALYSIS AND DESIGN:

Introduction

The proposed solution is a web application. The reason being the banks is filled with transactions of a wide range of fraud groups. The best and most effective way of detecting and preventing fraud is a good neural network model and a GUI for people to check the security of their card. In small-scale textile companies, a credit card is mostly predicted through manual methods. It is also wasted potential data for business intelligence. Most existing companies have one or more computers, so with a web application, there will not be a need to detect frauds manually an automated system will give an alert if any unusual transactions occur.

3.2.1 Requirement Analysis:

Stakeholder Identification

This can be used by both national and non-nationalized banks as this kind of application helps in both detection and prevention of frauds.

End Users:

- Direct users: The banks will be the first users of this application as it will help them in finding frauds and also preventing the frauds in an automatic process.
- Secondary users: The customers using this application as they have to keep their cards safe and also use their cards in a safe and sound manner

Beneficiaries:

The credit card users and also the banks will be benefited a lot from using this software as there is more transparency and also reduction of frauds.

3.2.2 Functional Requirements

1. The system will formulize the user's dynamic spending pattern based on first N purchases, comparing each purchase with the pattern and update the pattern gradually.
2. Many factors are considered in building the pattern including user's approximate annual income; gender; address; when and where most of the transaction had taken place; the average amount per transaction; the frequency of using ATM; user's tipping habits; categories of items that had been bought and so on.
3. In terms of comparing the most recent purchase P1 with the pattern P, the system will analyse a set of figures G, which includes the distance between P1 and P, the amount of transaction between P1 and P, the time of P1 and P and many other factors, used to come up with a numeric value Gs.

4. Based on the value of G_s , the system will determine the suspicious level S of P_1 and notify the users whether the credit card is fraud or not.

3.2.3 Non-Functional Requirements

1. **Reliability:** The project is guaranteed to provide reliable results for the entire user. The system shall operate 95% of the time. The number of defects 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.
2. **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. **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.
4. **Performance:** The response time for information to the user will be minimal and the user would get his desired results in a very short time.
5. **Accuracy:** The accuracy of the results obtained might not be very high at the initial stages as the number of datasets given for training the datasets is less. As the dataset given for training increases the accuracy of the model would also increase.
6. The file uploaded by the user should only contain the details about the credit card and other details won't be considered.

System Requirements

Hardware Requirements

- Processor – Intel i3 and above
- RAM: 4 GB and higher
- Hard Disk: 500GB minimum

Software Requirements

- Operating System: Windows or Linux
- Python IDE: python 2.7.x and above
- Visual Studios IDE required, jupyter notebook
- Setup tools and pip to be installed for 3.6 and above

- Language: Python scripting
- XAMPP Control Panel (PHP)
- Apache Server
- MySQL

Software Requirement Specification Document

The Development was done and tested with:

- Operation System: Windows 10
- Processor: 2.6 GHz Dual-Core Intel i7
- Memory: 12 GB 1600 MHz DDR3

Other dependencies:

Backend:

- PHP 4.16.2

PHP is a language that is particularly well suited to interact with databases. PHP can accept and validate the information that users type into a Web form and can also move the information into a database. PHP, a scripting language designed specifically for use on the Web, is a dynamic tool for creating dynamic Web pages. PHP is rich in features that make Web design and programming easier. Its popularity continues to grow, meaning that it fulfils its function pretty well.

- PYTHON 3.7.6
- TensorFlow 2.2.0
- Minsom 2.7.6
- MySQL:

Database is stored using SQL.

Frontend:

- HTML5:

Hypertext markup language Also called web pages A markup language is a set of markup. The Web pages created with HTML alone are static, meaning the user can't interact with the Web page. All users see the same Web page. Dynamic Web pages, on the other hand, allow

the user to interact with the Web page. Different users might see different Web pages. For instance, one user looking at a furniture store's online product catalog might choose to view information about the sofas, whereas another user might choose to view information about coffee tables.

- Bootstrap3:

It is a framework which used to create modern websites. It includes the features of HTML and CSS templates which is used in UI interface elements such as forms and buttons.

- CSS:

User to create the layout and look of pages

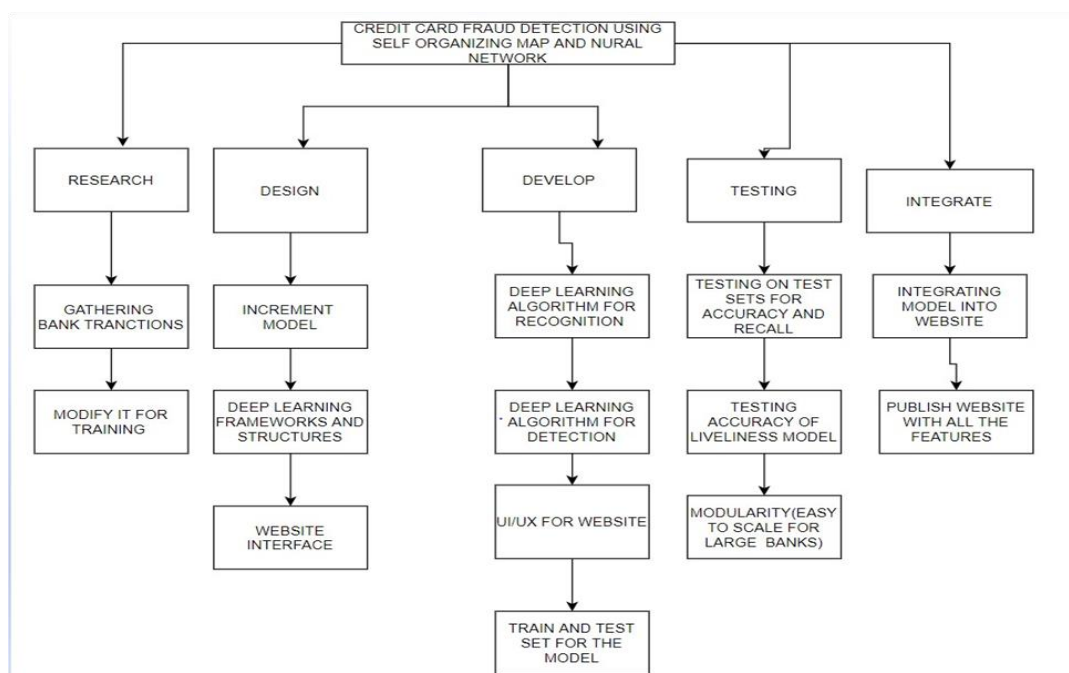
- JavaScript:

One language widely used to make Web pages dynamic is JavaScript. JavaScript is useful for several purposes, such as mouse-overs (for example, to highlight a navigation button when the user moves the mouse pointer over it) or accepting and validating information that users type into a Web form.

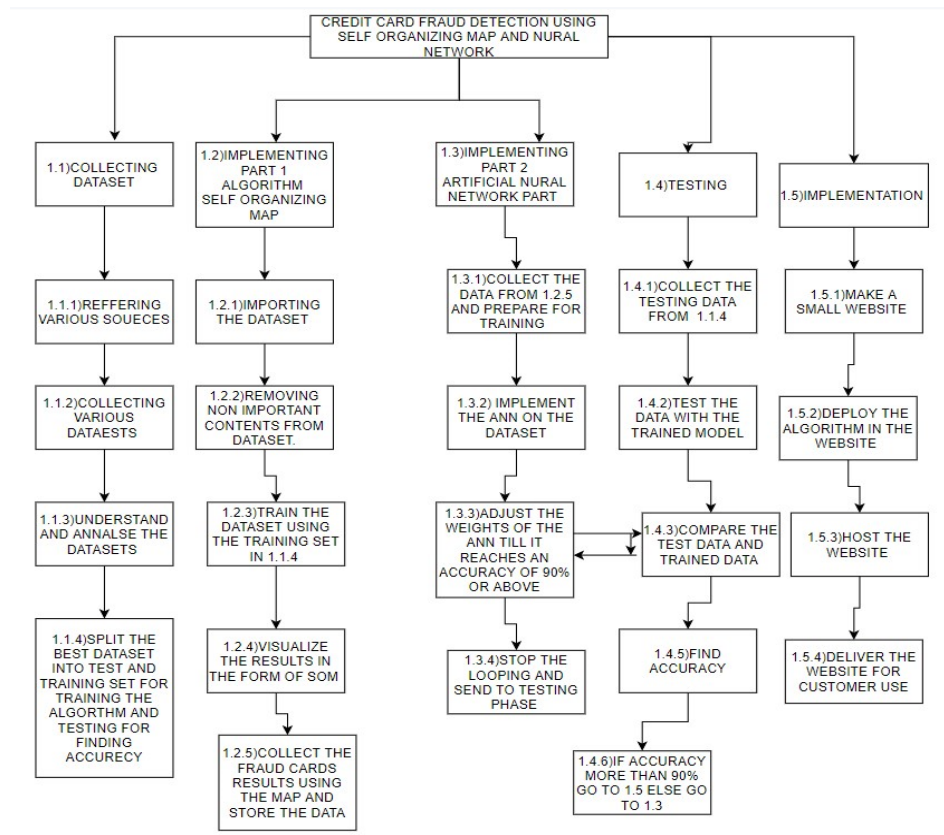
Browser: Version 86.0.4240.111 (Official Build) (x86_64)

Work Breakdown Structure

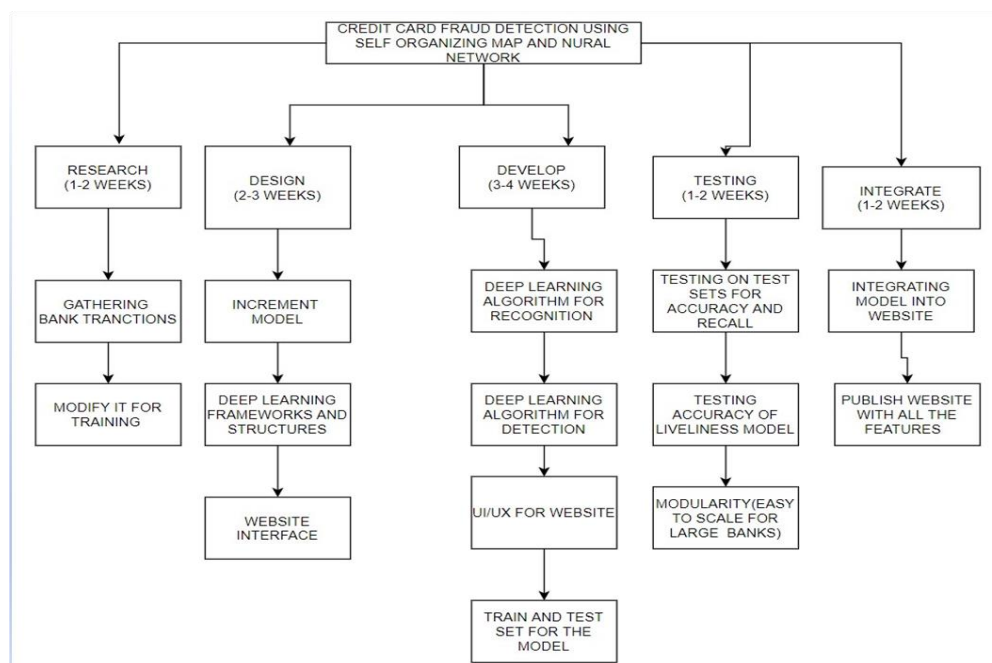
- Verb-Oriented WBS: A task-oriented WBS that defines the deliverable of the project work in terms of the actions that must be don't to produce the final product.



•Noun-oriented WBS: A deliverable-oriented WBS that defines the project work in terms of the components (physical and functional) that make up the deliverable.



•Time-Phased WBS: A time-oriented WBS that is used for very long-running projects to break the project into major phases instead of tasks.



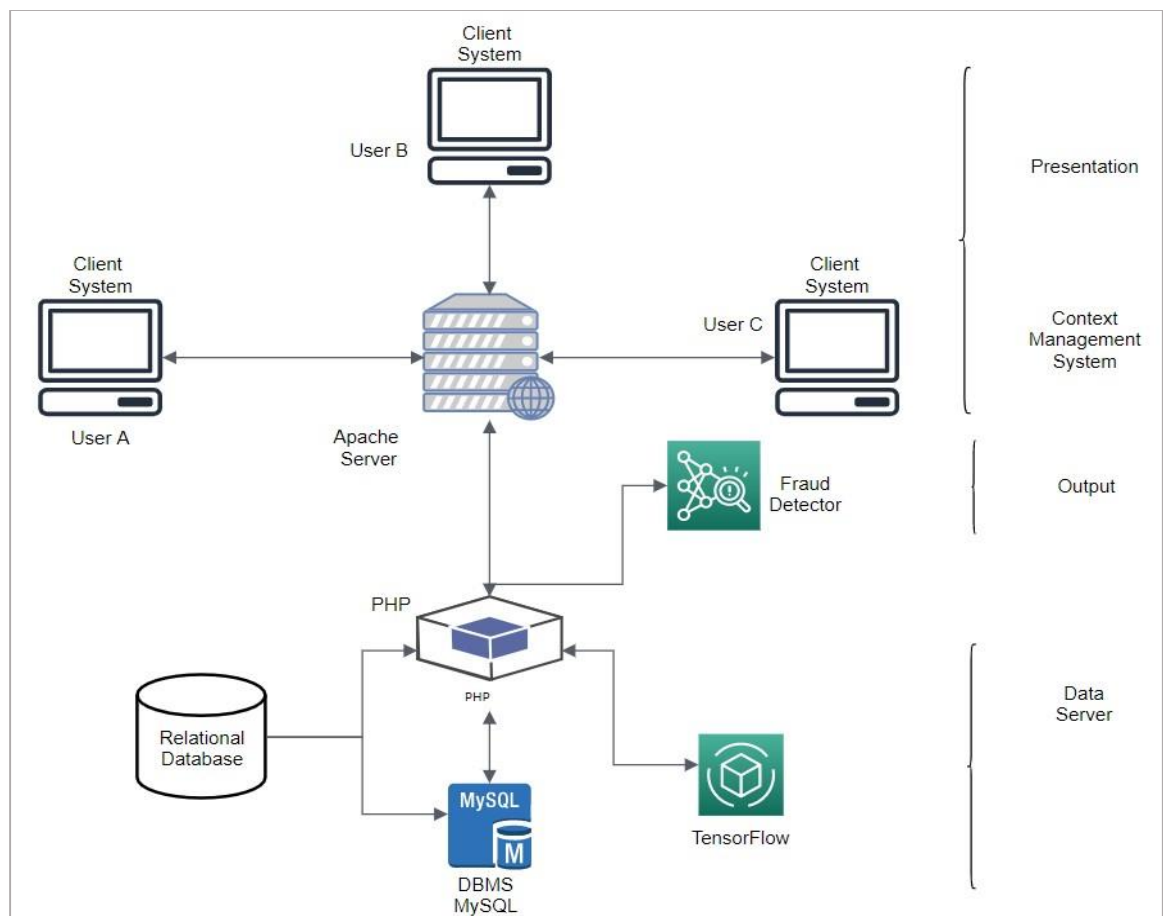
DESIGN OF PROPOSED SYSTEM:

Introduction

The factors considered in designing the Credit card fraud detection system are Interoperability and accessibility with minimum requirements on the user's side. Due to large flow of information delivery over the Internet, the system as a standard Internet application. The client side requires no more than standard Internet browser installed on the local computer, while the main application functionality is assured by the server side.

This includes, user interface made up of access services points at the remote site, a high speed, highly reliable and scalable regional network and content management gateway with database server. This architecture allows users to access the system via the Internet using hypertext transfer protocol and the user request is transformed into a structured query language using a PHP common content management gateway, which in turn passes it to the appropriate backend system. The common content management gateway provides a single point entry to the system.

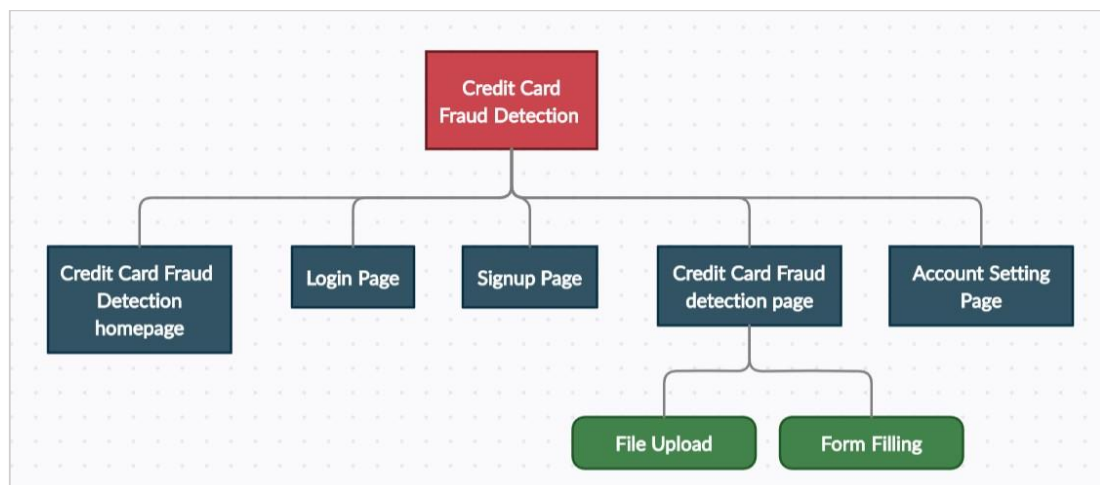
Architecture Diagram:



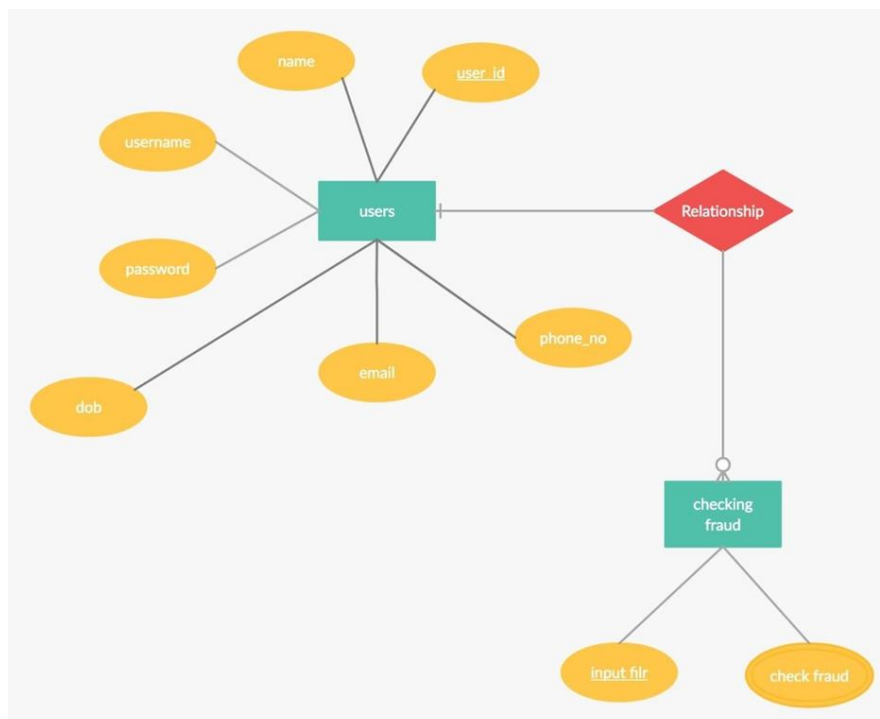
Architecture Diagram Explanation:

The web architecture is designed in such a way that we can use it as credit card fraud detection. The web app was developed using PHP, python as server-side scripting and also the website will help the user to input the file with his details that will be read by the server and based on the details it will determine whether the transactions will have fraud and also send an email to the user the probability of fraud that has occurred. We will also be using the TensorFlow module to perform the prediction and detecting the credit card frauds.

UI Design:



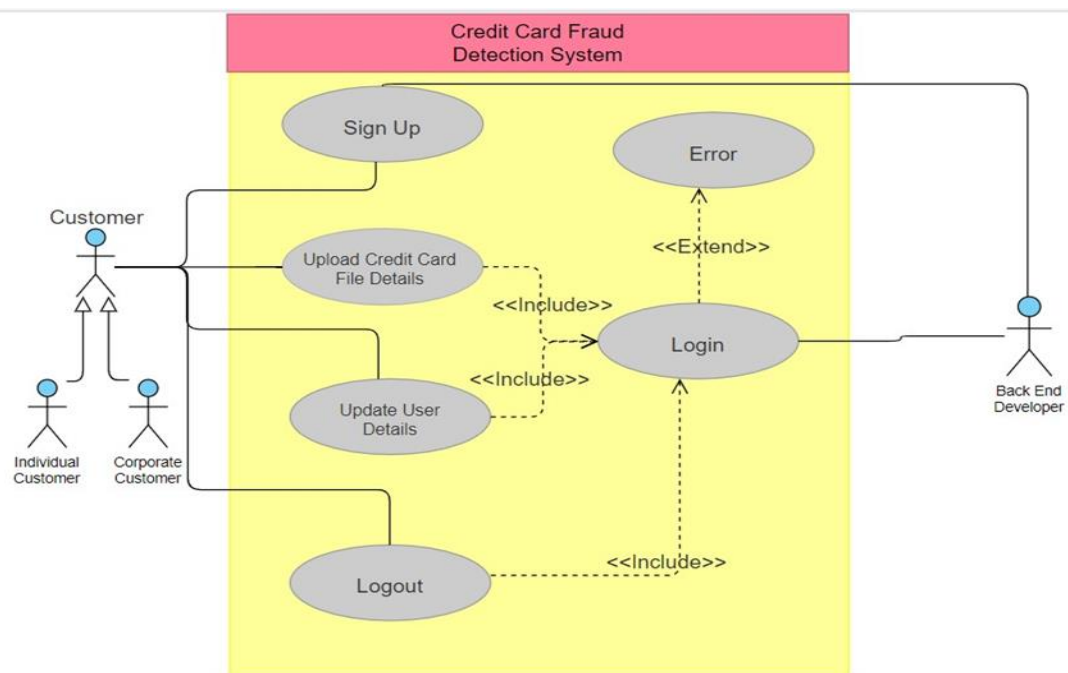
ER Diagram



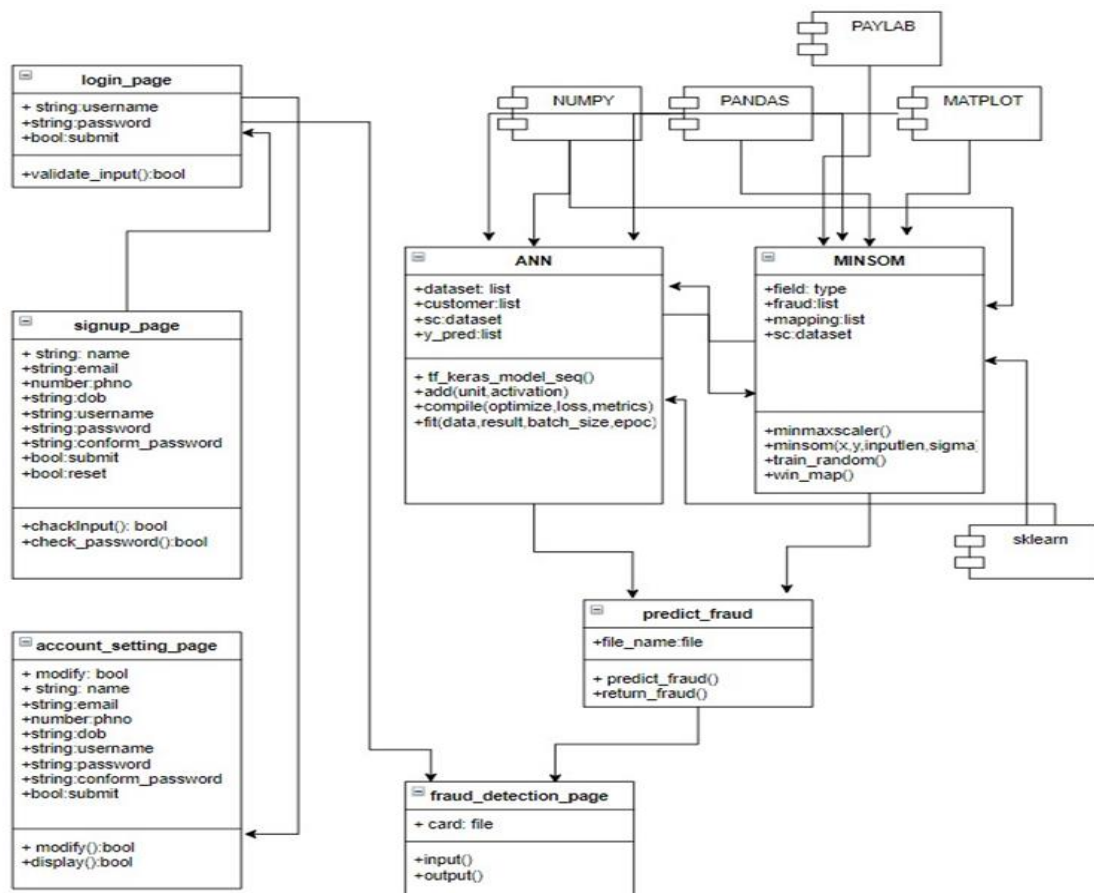
Use case Description in tabular format:

Use Case Title	Transfer Data
Description	A user who wants to find out whether his credit card is fraud or not, can enter his details to signup and login in which he can modify his account details or upload the details of his card in the form of file, which will inform the user whether his card is fraud or not.
Actor (s)	User, Back-end developer / Admin
Preconditions	The file must contain the details of card which is issued by the bank on request. The user must have an account in the website.
Postconditions	The user gets his result (credit card is fake or not) or his personal details that he has modified is updated.
Main success scenario	1.User enters his details and selects the “signup” button. 2.The user’s information’s are accepted and grants the user to login to the main page.
	3. User selects the “upload file” option and uploads his file containing the card details. 4. The user gets his desired results. 5. Or the user selects “account setting” option and modifies his personal information. 6. His account details have been updated.
Extensions	1.The user does give proper personal information. 1.1 The user’s account is not created. 1.2 User backs out of the use case 2.The user does not upload file with proper card details 2.1 User’s file is not getting accepted 2.2 User doesn’t get his desired results.

UML diagram:



Class Diagram:

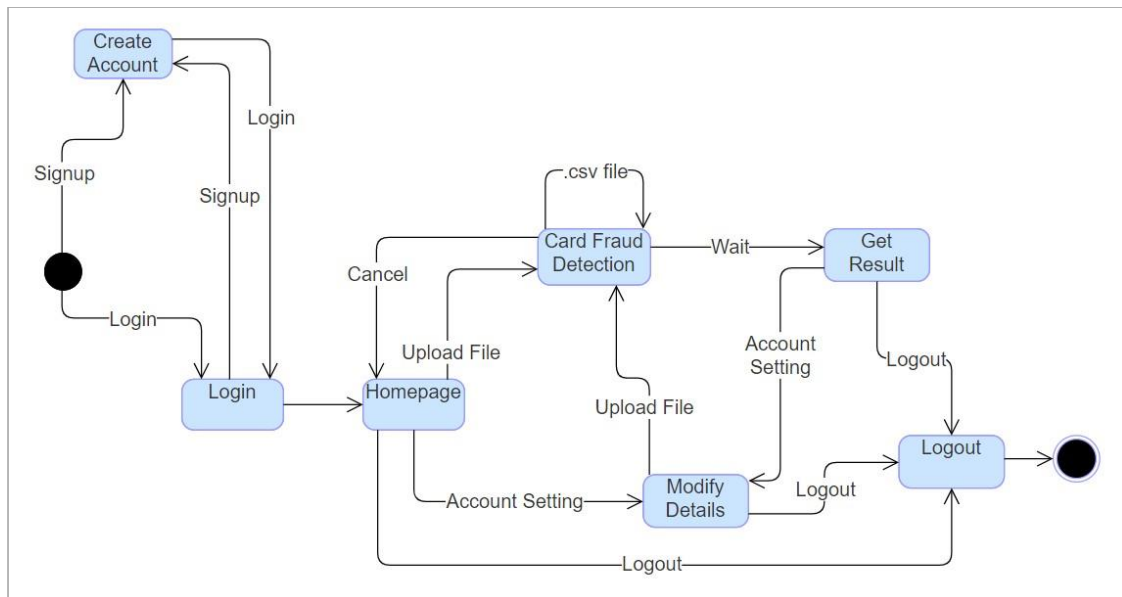


State chart Diagram State Description:

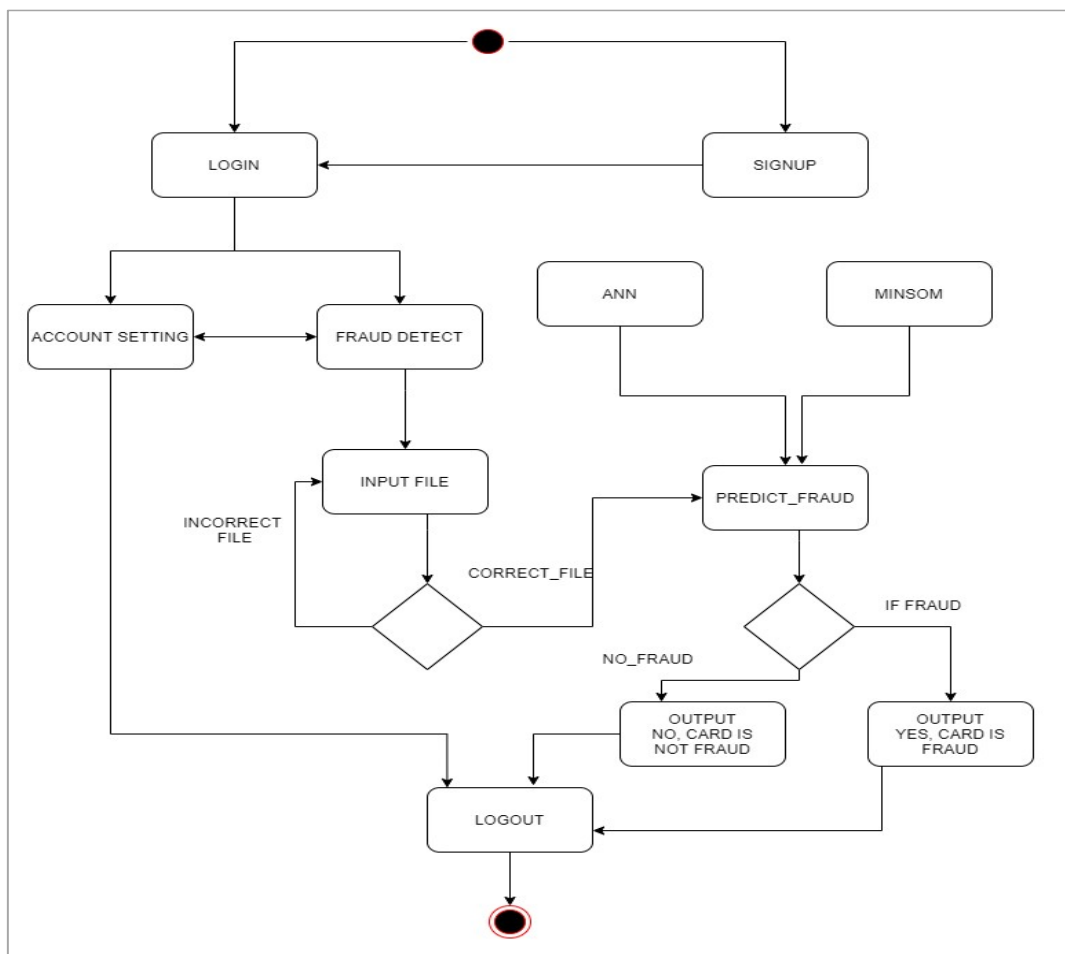
State	Description
Create Account	The user has created his / her count. Monitor shows 'account created successfully' and redirects to Login page
Login	The user has logged in to his account. Monitor shows 'successfully logged in' and redirects to the file upload page.
Homepage	The user is in his / her homepage. Monitor shows the homepage with option to upload the file and account setting.
Card Fraud Detection	The user has uploaded the file to determine whether his card is fraud or not. Monitor shows 'evaluating your details'.
Modify Details	The user has modified his / her account details. Monitor shows 'account updated'.
Get Result	The user gets the result for the file he / she has uploaded. Monitor shows 'card is fraud or not'.
Logout	The user has logged out of his / her account. Monitor shows 'Thank you for using'.

Stimulus Description:

Stimulus	Description
Signup	The user has pressed the Sign-up button.
Login	The user has pressed the Login button.
Upload File	The user has pressed the Upload File button.
Cancel	The user has pressed the Cancel button.
Account Setting	The user has pressed the Account setting button.
.csv file	The user has uploaded a CSV file.
Wait	The user has uploaded the file and is waiting for the results.
Logout	The user has pressed the Logout button.



Activity Diagram:

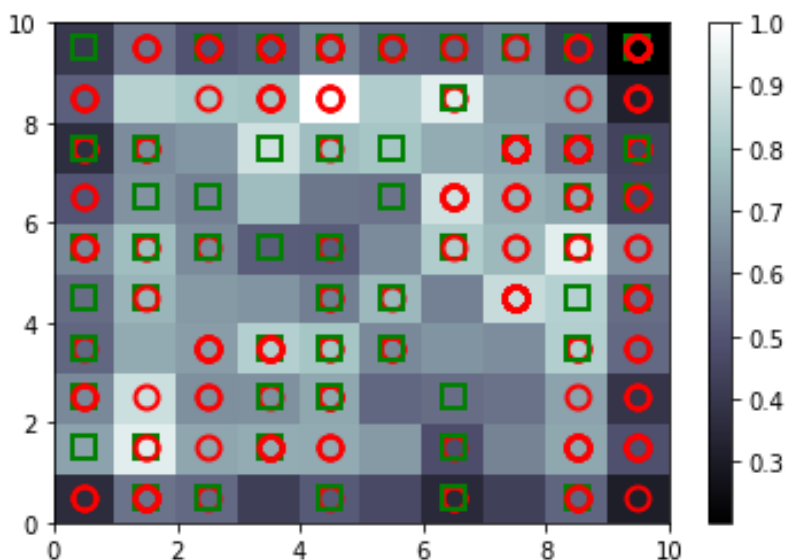


IMPLEMENTATION AND TESTING:

The initial code to plot the dataset against SOM

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Mon Mar 22 17:48:43 2021
4
5  @author: vignesh
6  """
7
8  # # Importing the libraries
9  import numpy as np
10 import pandas as pd
11 import matplotlib.pyplot as plt
12 # # Importing the dataset
13 dataset = pd.read_csv('C:/Users/vignesh/Desktop/Credit_Card_Applications.csv')
14 X = dataset.iloc[:, :-1].values
15 y = dataset.iloc[:, -1].values
16 from sklearn.preprocessing import MinMaxScaler
17 sc = MinMaxScaler(feature_range = (0,1))
18 X = sc.fit_transform(X)
19 from minisom import MiniSom
20 som = MiniSom(x=10, y=10, input_len= 15, sigma= 1.0,
21 learning_rate = 0.5)
22 som.random_weights_init(X)
23 som.train_random(data = X, num_iteration = 100)
24 from pylab import bone, pcolor, colorbar, plot, show
25 bone()
26 8
27 pcolor(som.distance_map().T)
28 colorbar()
29 markers = ['o', 's']
30 colors = ['r', 'g']
31 for i, x in enumerate(X):
32     w = som.winner(x)
33     plot(w[0] + 0.5,
34          w[1] + 0.5,
35          markers[y[i]],
36          markeredgecolor = colors[y[i]],
37          markerfacecolor = 'None',
38          markersize = 10,
```

The outliers are then plotted using the SOM map



The code for plotting ANN:

```

43 print('Fraud Customer IDs')
44 for i in frauds[:, 0]:
45     print(int(i))
46
47 #creating features
48 customers = dataset.iloc[:, 1:].values
49
50 #creating a dependant variable
51 is_fraud = np.zeros(len(dataset))
52 for i in range(len(dataset)):
53     if dataset.iloc[i,0] in frauds:
54         is_fraud[i] = 1
55
56 #feature scaling for ann
57 from sklearn.preprocessing import StandardScaler
58 sc = StandardScaler()
59 customers = sc.fit_transform(customers)
60
61 #making the ANN
62
63 #importing keras
64 from keras.models import Sequential
65 from keras.layers import Dense
66
67 #initialising the ANN
68 classifier = Sequential()
69
70 #adding the input layer and hidden layer
71 classifier.add(Dense(units = 2, kernel_initializer = 'uniform', activation = 'relu', input_dim = 15))
72
73 #adding the output layer
74 classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
75
76 #compiling the ANN
77 classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
78
79 #fitting the ANN to training set
80 classifier.fit(customers, is_fraud, batch_size = 1, epochs = 2)
81
82 #predicting the probabilities
83 y_pred = classifier.predict(customers)
84 v_pred = np.concatenate((dataset.iloc[:, 0:1].values, y_pred), axis = 1)

```

Variable explorer:

Name	Type	Size	Value
colors	list	2	['r', 'g']
customers	Array of float64	(690, 15)	[[0.68873723 -0.80185183 ... -0 ...
dataset	DataFrame	(690, 16)	Column names: CustomerID, A...
frauds	Array of float64	(8, 15)	[[1.5757467e+07 0.0000000e+... 0.0000 ...
i	int	1	689
is_fraud	Array of float64	(690,)	[0. 0. 0. ... 0. 0. 0.]
mapoines	defaultdict	74	defaultdict object of

Console 2/A:

```

...: frauds = np.concatenate((mappings[(4,0)], mappings[(4,5)]), axis = 0)
...: frauds = sc.inverse_transform(frauds)

In [7]: print('Fraud Customer IDs')
...: for i in frauds[:, 0]:
...:     print(int(i))
...:
Fraud Customer IDs
15757467
15815443
15748432
15699963
15648876
15696287
15698749
15667468

In [8]: customers = dataset.iloc[:, 1:].values

```

Next up we have executed the code in Jupyter Notebook to determine the accuracy and loss:

```

[19] In: #Training the ANN

[20] In: #Compiling the ANN
ann.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

[21] In: #Training the ANN on the Training set
ann.fit(customers, is_fraud, batch_size = 1, epochs = 10)

Epoch 1/10
690/690 [=====] - 1s 1ms/step - loss: 0.5324 - accuracy: 0.8246
Epoch 2/10
690/690 [=====] - 1s 1ms/step - loss: 0.3598 - accuracy: 0.9319
Epoch 3/10
690/690 [=====] - 1s 2ms/step - loss: 0.2989 - accuracy: 0.9319
Epoch 4/10
690/690 [=====] - 1s 2ms/step - loss: 0.2696 - accuracy: 0.9319
Epoch 5/10
690/690 [=====] - 1s 2ms/step - loss: 0.2528 - accuracy: 0.9319
Epoch 6/10
690/690 [=====] - 1s 1ms/step - loss: 0.2417 - accuracy: 0.9319
Epoch 7/10
690/690 [=====] - 1s 2ms/step - loss: 0.2347 - accuracy: 0.9319
Epoch 8/10
690/690 [=====] - 1s 1ms/step - loss: 0.2287 - accuracy: 0.9319
Epoch 9/10
690/690 [=====] - 1s 2ms/step - loss: 0.2211 - accuracy: 0.9319
Epoch 10/10
690/690 [=====] - 1s 2ms/step - loss: 0.2122 - accuracy: 0.9319
<tensorflow.python.keras.callbacks.History at 0x14e241248>

[22] In: #Predicting test set results
y_pred = ann.predict(customers)
y_pred = np.concatenate((dataset.iloc[:, 0:1].values, y_pred), axis = 1)
y_pred = y_pred[y_pred[:, 1].argsort()]

[30] In: #copying values to scale
new_list=y_pred.copy()

[31] In: #printing and checking the new list

```

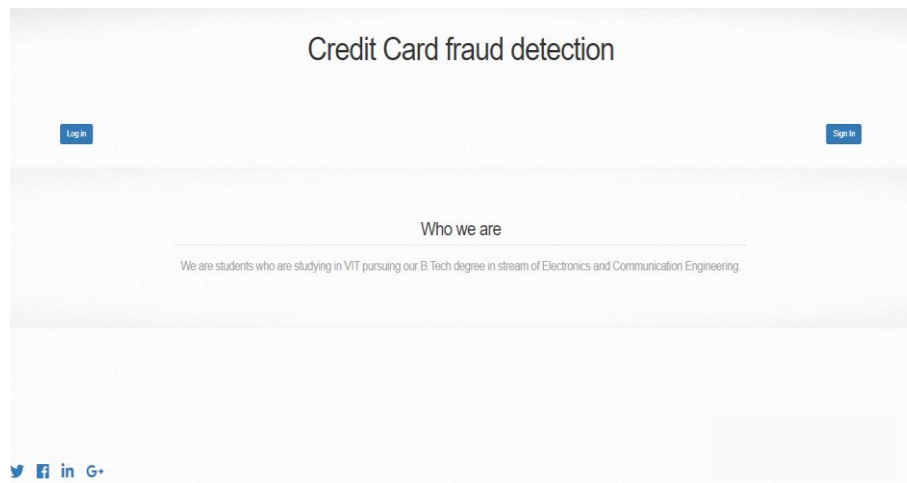
The probabilities are then transferred over to a csv sheet

A	B	C
credit_card_id	probability of fraud	
15651460	0.000452872	
15654625	0.001686214	
15767264	0.002014948	
15647295	0.002146249	
15667934	0.002225659	
15719940	0.00325755	
15672637	0.003465641	
15706762	0.004081787	
15651868	0.004110529	
15736533	0.004272755	
15615296	0.004372258	
15801072	0.004418122	
15736399	0.004437865	
15766734	0.004768259	
15779207	0.005856147	
15769548	0.006754815	
15656417	0.006791593	
15750545	0.007658462	
15672894	0.007664228	
15633944	0.007667373	
15761733	0.007706248	
15615176	0.007730033	

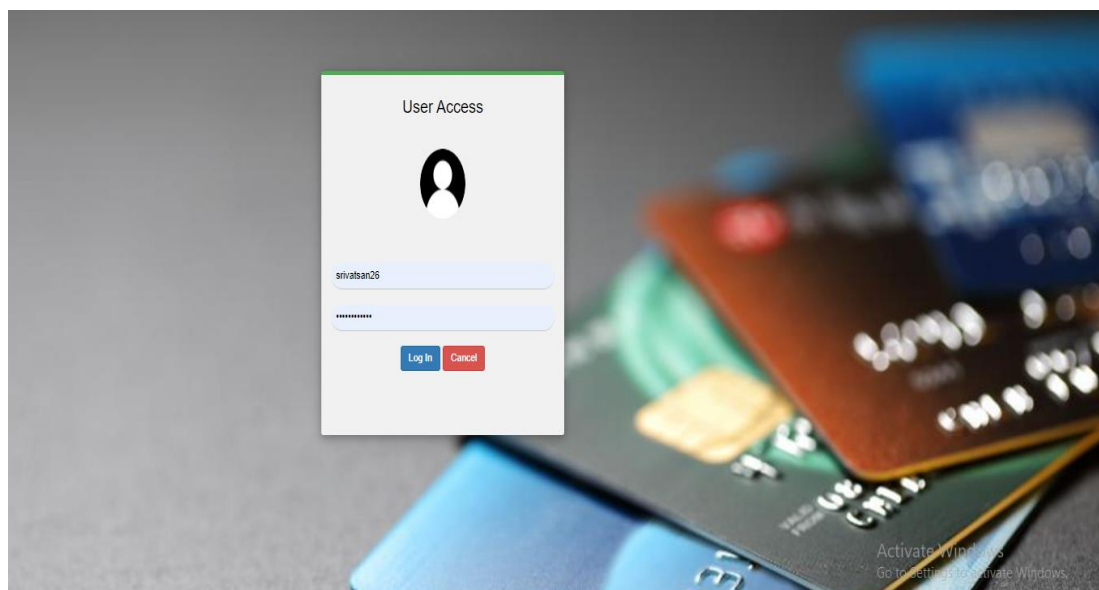
A	B	C
credit_card_id	probability of fraud	
15583680	0.485011548	
15611189	0.485011548	
15712877	0.486107037	
15646563	0.488296616	
15750055	0.488731143	
15679801	0.489083639	
15732943	0.489433164	
15762716	0.492490573	
15807546	0.494796603	
15704081	0.499893683	
15587038	0.500570458	
15762045	0.501567318	
15789014	0.503723788	
15592326	0.503918338	
15805212	0.504122148	
15775235	0.504477264	
15571284	0.506228211	
15578722	0.508178339	
15770255	0.511382687	
15728082	0.51380936	
15752601	0.516085775	
15812766	0.520370761	
15660097	0.530176506	

A	B	C
credit_card_id	probability of fraud	
15706602	0.555304096	
15773869	0.556444051	
15607986	0.556710759	
15600027	0.557788339	
15691150	0.558794547	
15716276	0.564365203	
15731989	0.565205427	
15649101	0.570922061	
15717629	0.574488951	
15762392	0.575050673	
15700511	0.580671211	
15678210	0.583836335	
15588123	0.587921617	
15598574	0.593609247	
15575243	0.597146086	
15803682	0.606389949	
15674583	0.613625765	
15794868	0.617043708	
15766183	0.617069042	
15707681	0.619174232	
15801817	0.627300854	
15664615	0.627598663	
15707146	0.630765005	

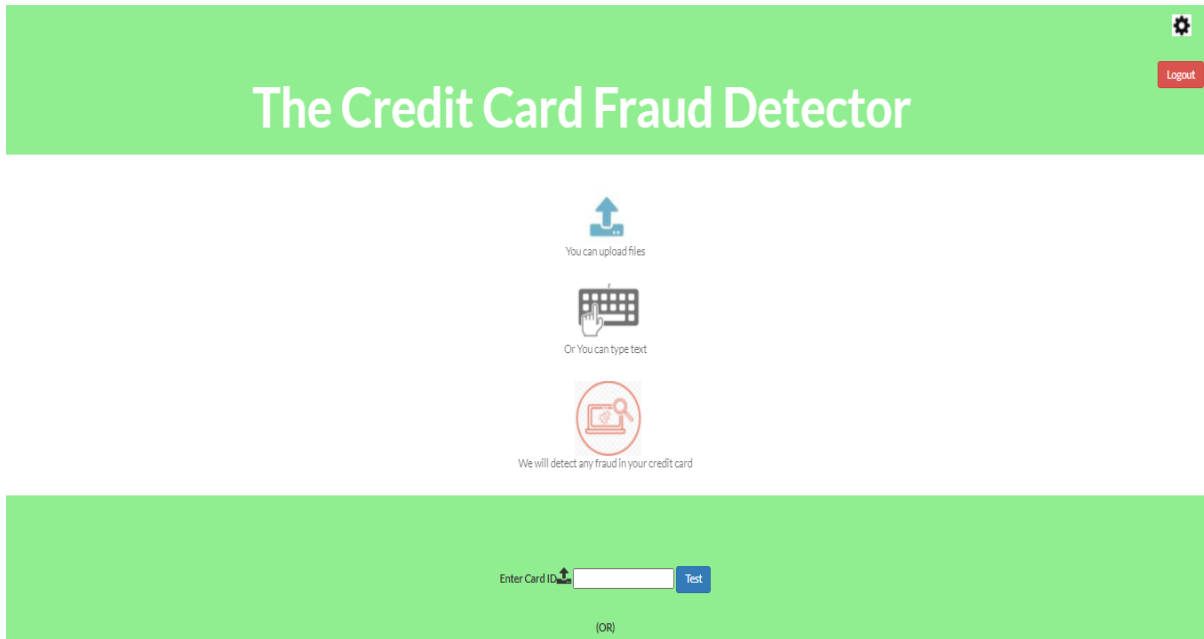
Our website index page:



The website login page:

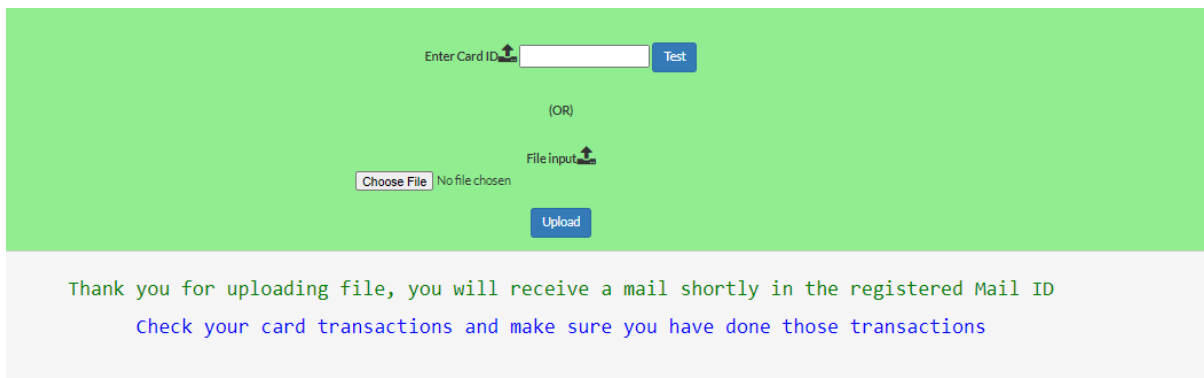


Main webpage:

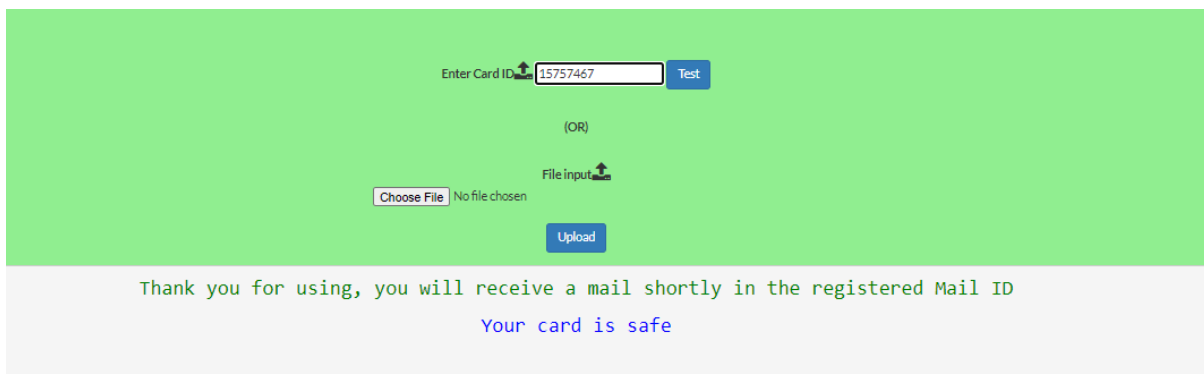


The screenshot shows the main webpage of 'The Credit Card Fraud Detector'. The header is green with the title 'The Credit Card Fraud Detector' in white. A 'Logout' button is in the top right. The main content area is white and contains three icons with text: 'You can upload files' (upload icon), 'Or You can type text' (keyboard icon), and 'We will detect any fraud in your credit card' (credit card icon). Below this is a green section with a form: 'Enter Card ID' with an upload icon, a text input field, a 'Test' button, and '(OR)' below it.

Few example responses of test cases:



This screenshot shows a test case response. The top section is green and contains the 'Enter Card ID' form with the 'Test' button. Below it is '(OR)' and the 'File input' section with a 'Choose File' button, 'No file chosen' text, and an 'Upload' button. The bottom section is light gray and contains the message: 'Thank you for uploading file, you will receive a mail shortly in the registered Mail ID' and 'Check your card transactions and make sure you have done those transactions'.



This screenshot shows another test case response. The top section is green and contains the 'Enter Card ID' form with the card ID '15757467' entered and the 'Test' button. Below it is '(OR)' and the 'File input' section with a 'Choose File' button, 'No file chosen' text, and an 'Upload' button. The bottom section is light gray and contains the message: 'Thank you for using, you will receive a mail shortly in the registered Mail ID' and 'Your card is safe'.

CONCLUSION, LIMITATION AND FUTURE WORK:

The described method using Artificial Neural Networks and SOM has better chances of successfully detecting credit card extortion. With every parameter verified and well represented in plot diagram. The singularity of our approach lies in using the clustering mechanism of SOM for the detection of credit card extortion activities. This helps in detecting hidden patterns of the transactions which can't be identified when compared with the other common methods used as of now. With appropriate colour schemes over representation of data sets and with the help of thousands of iterations the system is trained and then it can be used to predict the chances of fraud in the next transaction. Therefore, concept of SOM will be extremely efficient in the detection of the anomalies in credit card fraud cases. Hence, this model may help user or bank to manage frauds and to avoid the occurrence of fraud as early as possible.

The Future Expectations of the project are very vast especially with the case of AI. As the role of Artificial Intelligence is increasing rapidly, it can be of great use if the AI is used extensively in this field. The AI may help to deduce more efficient way of understanding the datasets and also to categorize the Transactions that helps in the better prediction of possibilities in the Transaction. The implementation of the AI will not only help in the detection of frauds but it may also help to deduce the various possibilities that may affect the transaction or underline the set of companies or individuals which have been related to the fraudulent transactions that occurred in the past. It may analyse the data and learn on its own about the factors which may further lead to a fraud and thus immediately instructs about it to the user.

REFERENCES:

- [1] A. Vellidoa, P.J.G. Lisboaa, J. Vaughan “Neural networks in business: a survey of applications”. Elsevier, Expert Systems with Applications, (1999). 17; (51–70).
- [2] R. T. Trippi, E. Turban (eds), Neural Networks in Finance and Investing, Probus Publishing Company (1993).
- [3] A.J. Graaff A.P. Engelbrecht “The Artificial Immune System for Fraud Detection in the Telecommunications Environment”; (2011). (1-4)
- [4] Aihua Shen, Rencheng Tong, Yaochen Deng “Application of Classification Models on Credit Card Fraud Detection”. (2007).
- [5] Anshul Singh, Devesh Narayan “A Survey on Hidden Markov Model for Credit Card Fraud Detection”. International Journal of Engineering and Advanced Technology (IJEAT), (2012). Volume-1, Issue-3; (49- 52).

- [6] B.Sanjaya Gandhi , R.Lalu Naik, S.Gopi Krishna, K.lakshminadh “Markova Scheme for Credit Card Fraud Detection”. International Conference on Advanced Computing, Communication and Networks; (2011). (144- 147).
- [7] Bidgoli, B. M., Kashy, D., Kortemeyer, G. & Punch, W. F “Predicting student performance: An Application of data mining methods with the educational web-based system LON-CAPA”. In Proceedings of ASEE/IEEE frontiers in education conference. . (2003).
- [8] Bolton, R. J., Hand, D. J (2002). “Statistical fraud detection: A review”. Statistical Science (1994).28(3); (235—255).
- [9] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler “A comprehensive survey of data mining-based fraud detection research”. In Artificial Intelligence Review. (2005).
- [10] Cortes, C. & Vapnik, V “Support vector networks, Machine Learning”. . (1995). Vol. 20; (273–297).
- [11] De Castro Silva, L. N., & Zuben, F. J. V “An evolutionary immune network for data clustering”. In Proceedings of the IEEE SBRN (Brazilian Symposium on Artificial Neural Networks); . (2000). (84–89).
- [12] De Castro, L., & Timmis, J “Artificial immune systems: a new computational approach”. London, UK: SpringerVerlag. . (2002)
- [13] Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, Survey of Fraud Detection Techniques, Proc. IEEE : International Conference on Networking , Sensing & Control, pp. 749-754, 2004.
- [14] [Jeske, D.R. Lin, P.J. Rendon, C. Rui Xiao Samadi, B., Synthetic Data Generation Capabilities for Testing Data Mining Tools. IEEE : Military Communications Conference, pp.1-6, 2006
- [15] Abhinav Srivastava, Amlan Kundu, Shamik Sural, Arun K. Majumdar. “Credit Card Fraud Detection using Hidden Markov Model”. IEEE Transactions on dependable and secure computing, Volume 5; (2008) (37- 48)