Credit Card Fraud Detection Using SVM

A project report submitted in the partial fulfillment of the requirements for the Award of the Degree of

MASTER OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

V.RAKJYA LAKSHMI	**17811A0550**
K.BHARGAV	**17811A0528**
U.KEERTHI	**17811A0548**
Y.YASWANTH KUMAR	**17811A0552**

Under the esteemed guidance of

Sri . V. TRINADH

Associate Professor

(Department Of Computer Science&Engineering)
A.I.E.T

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Affiliated to JNTU ,Kakinada & NBA Accredited)

Makavarapalem, Narsipatnam-531113

2017-2021

AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(NBA Accredited)

(Affiliated to Jawaharlal Nehru Technological University-Kakinada)

MAKAVARAPALEM, VISAKHAPATNAM-531113

DEPARTMENT OF COMPUTER SCIENCE &ENGINEERING

CERTIFICATE

This is to certify that this project work entitled **"CREDIT CARD FRUAD DETECTION USING SVM"** is bonafide record work done by **V.RAJYA LAKSHMI (17811A0550)** students of final year B.Tech in the department of Computer Science&Engineering in **Avanthi Institute Of Engineering &Technology,Visakhapatnam**. This work was done for the partial fulfillment for the requirement for the Award of Bachelor of Technology during the **2017-2021** academic years.

The results submitted in this project have been verified and are found to be satisfactory. The results embodied in this thesis have not been submitted to any other university for the award of the any other degree/diploma.

Sri. V.TRINADH
Asst.professor
Project Guide

Sri.U.NANAJI
Head of theDepartment,
Dept.of Computer
Science Engg.

External Examiner

ACKNOWLEDGEMENT

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

The first person I would like to thank is my project guide **V.TRINADH, Associate Professor**, Department of Computer Science and engineering, had given continuous critical suggestions and extension of proper working atmosphere, abiding interest has finally evolved into this research work.

It is indeed with a great sense of pleasure and immense sense of guidance that I acknowledge the help and I am highly indebted to our principal **Dr.C.P.V.N.J.Mohan Rao** provided to accomplish this work.

I would like to express my sincere thanks to **U.NANAJI** Department of Computer Science and Engineering.

I owe special thanks to Departmental committee for the successful completion of this project. I am grateful to all staff members, Department of Computer Science and Engineering, xxxxxx for their valuable suggestions and encouragement.

Project Associates

V.RAJYA LAKSHMI	**(17811A0550)**
K.BHARGAV	**(17811A0528)**
U.KEERTHI	**(17811A0548)**
Y.YASWANTH KUUMAR	**(17811A0552)**

LIST OF CONTENTS

Page no

Abstract

In day to day life credit cards are used for purchasing goods and services with the help of virtual card for online transaction or physical card for offline transaction. In a physical-card based purchase, the cardholder presents his card physically to a merchant for making a payment. To carry out fraudulent transactions in this kind of purchase; an attacker has to steal the credit card. If the cardholder does not realize the loss of card, it can lead to a substantial financial loss to the credit card company.

In online payment mode, attackers need only little information for doing fraudulent transaction (secure code, card number, expiration date etc.). In this purchase method, mainly transactions will be done through Internet or telephone. To commit fraud in these types of purchases, a fraudster simply needs to know the card details. Most of the time, the genuine cardholder is not aware that someone else has seen or stolen his card information. The only way to detect this kind of fraud is to analyze the spending patterns on every card and to figure out any inconsistency with respect to the “usual” spending patterns.

1. INTRODUCTION

1.1 INTRODUCTION:

The development of economic and the open financial market make the credit card business become one of the bank's most important incomes. But along with the growth of issuance volume, global credit fraud transactions increase at an alarming rate. Financial companies cannot effectively discover fraudulent transactions; as a consequence the loss is becoming increasingly serious. How to identify the credit card fraudulent transactions effectively, quickly and accurately is becoming the generally concerned problem. In China, we begin to use a credit card of payment online in recent years. The related study is divided into two directions: fraudulent identification and enterprise applications.

The researches of the first direction are Tong Fengru's which is based on the combination of classifier and Yan Hua, Hu Mengliang's which is using Bayesian classification algorithm. In the latter direction, the third party payment merchant—IPS officially released a credit card anti-fraud system called ANT [1]. The system uses a neural network-based anti-fraud model for parallel docking with the credit card payment instruments, effective inhibition of the credit card electronic payment to the various risks that may occur during the transactions. Early research of credit card fraud prevention focuses on the classification and identification of methods and models, including single pattern recognition methods such as decision trees and neural networks, the combination method, distributed data mining. However due to the complexity and sparse of the transaction data, these methods are often faced with the issue of model selection, model parameter settings, improper selection when dealing with large scale transaction data, which often lead to owe study, over fitting and the local optimalproblem[2]. Support vector machine is a relatively new field in the data mining field. The process is first mapped data from the input space to feature space, and then construct a linear discriminate function in feature space. Although there are many similarities between neural network and support vector machine in structure, the latter is relatively simple.

Applications:

Financial market makes the credit card business become one of the bank’s most important incomes. But along with the growth of issuance volume, global credit fraud transactions increase at an alarming rate. Financial companies cannot effectively discover fraudulent transactions.

Motivation:

In order to identify the credit card fraudulent transactions, in this paper we propose an optimized SVM model for detection of fraudulent online credit card model.

Problem Statement:

Payments using credit cards have increased in recent years. It may be used in online or in regular shopping. Now-a-days credit card payments are necessary and convenient to use. Due to the increase of fraudulent transactions, there is a need to find the efficient fraud detection model.

Objectives:

1.To Propose a new Credit Card Fraud Detection based on Data Mining using Support VectorMachines

2.To employ the incremental learning technique to reduce the misclassification rate and generation of false alarms 4.To Evaluate the proposed technique using various input and output parameters such as Classification errors, Accuracy and FalseAlarms.

SOFTWARE REQUIREMENT SPECIFICATION

Requirements Specification:

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

HARDWARE REQUIREMENTS:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

➢ System	: Pentium IV 2.4 GHz.
➢ Hard Disk	: 100 GB.
➢ Monitor	: 15 VGA Color.
➢ Mouse	: Logitech.
➢ RAM	: 1 GB.

SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

➢ Operating system	:	Windows XP/7/10
➢ Coding Language	:	python
➢ Tool	:	Jupiter
➢ IDE	:	Anaconda prompt

FUNCTIONAL REQUIREMENTS:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes.
Modules

- **Dataset collection and loading:**

  Using this module we will collect dataset from kaggle website which has features and labels denoted as v1 to v60 as features and 1,0 as labels. This data set is in csv format this is loaded

- **Preprocessing:**

  In this module dataset is organized in required manner and data set features and labels are stored in two variables.

- **Data set Testing and Training:**

  Uisngsklearn library using train and test function data set is divided in to train, test where train will have x, y values features and labels of all records in dataset where as in test only half of records will be there**.**

- **Analysis and Prediction accuracy checking:**

  In this module test and train data is analyzed and data is shown in the form of graphs and svm algorithm is used to predict results using test data with train data.

NON FUNCTIONAL REQUIREMENTS :

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

- Accessibility
- Availability
- Backup
- Certification
- Compliance
- Configuration Management
- Documentation
- Disaster Recovery
- Efficiency(resource consumption for given load)
- Interoperability

PERFORMANCEREQUIREMENTS:

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the
other hand designing a system, which does not cater to the requirements of the user, is of no use.
The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

Feasibility Study

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

Operational Feasibility

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc

Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses sqllite, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

Economic Feasibility

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different

LITERATURE SURVEY:

There was a lot of research work carried out for credit card fraud detection. CARDWATCH, a data base mining system proposed by Aleskerov et al. [7]. It was based on neural networks. In this model, customers past transactions are trained in the neural network. Then the network checks the current spending pattern with the past data, if deviations appear then it is considered as suspicious. Zhang Yongbin et al. [8] suggested a behavior based credit card fraud detection model. Here they use the historical behavior pattern of the customer to detect the fraud. The transaction record of a single credit card is used to build the model. In this model, unsupervised Self organizing map method is used to detect the outliers from the normal ones.

Chuang et al. [9] developed a model based on data mining. They used the web services to exchange data between banks and fraud pattern mining algorithm for detection. With the proposed scheme participant banks can share the knowledge about fraud patterns in a heterogeneous and distributed environment and further enhance their fraud detection capability and reduce financial loss.

Wen-Fang Yu et al. [10] proposed an outlier mining method to detect the credit card frauds. Definitions of Distance based outliers are referred and the outlier mining algorithm was created. This model detects outlier sets by computing distance and setting threshold of outliers. It efficiently detects the overdrafts and is also used to predict the fraudulent transactions.

Tao Guo et al. [11] applied the neural data mining method. This model is based on customer’s behavior pattern. Deviation from the usual behavior pattern is taken as an important task to create this model. The neural network is trained with the data and the confidence value is calculated. The credit card transaction with low confidence value is not accepted by the trained neural network and it is considered as fraudulent. If the confidence value is abnormal, then again it is checked for additional confirmation. The detection performance is based on the setting of threshold. SuvasiniPanigrahi et al. [12] suggested a fusion approach. It consists of four components namely, rule based filter, DempsterShafer Adder, transaction history database and Bayesian learner. Rule

based filter is used to find the suspicion level of the transaction. Dempster-Shafer Theory is used to compute the initial belief which is based on the evidences given by the rule based filter. The transactions are classified as normal, abnormal or suspicious depending on this initial belief. Once a transaction is found to be suspicious, belief is further strengthened orweakened according to its similarity with fraudulent or genuine transactions history using Bayesian learning.

Abhinav Srivastava et al. [13] developed the hidden Markov model (HMM) to detect the credit card fraud. An HMM is initially trained with the normal behavior of the cardholder. If the current transaction is not accepted by the trained HMM with high probability, it is considered to be fraudulent. Vladimir et al. [14] applied self organizing map algorithm to create a fraud detection model. The pattern of legal and fraudulent transactions is observed from the earlier transactions and it is created based on the neural network training. If a new transaction does not match to the pattern of legal cardholder or is similar to the fraudulent pattern it is classified as suspicious for fraud. Chen et al. [15] proposed the online questionnaire method to collect the transaction data of the users. A SVM is trained with the data and the questionnaire responded transaction is used to predict the new transactions

.

System Design

SYSTEM ARCHITECTURE

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It's a serious impact on the later part, notably testing and maintenance.

The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

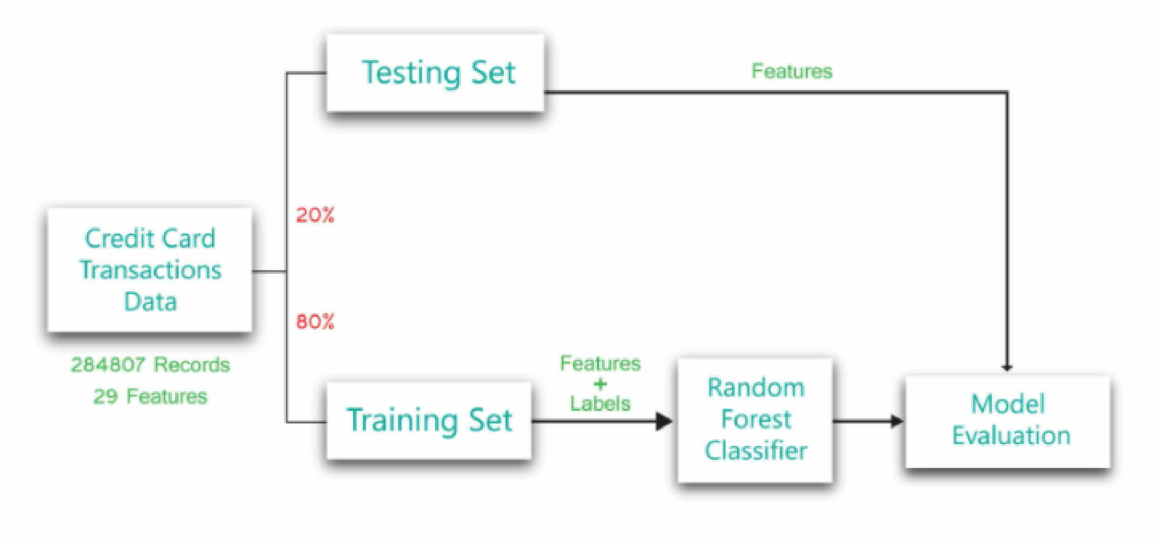


Figure 7.1: Architecture diagram\

Training set: a set of examples used for learning: to fit the parameters of the classifier In the SVM case, we would use the training set to find the “optimal” Support Vectors

Validation set: a set of examples used to tune the parameters of a classifier For SVM case, we would use the validation set to find the “optimal” number of support vectors or determine a stopping point for the algorithm

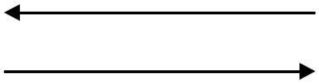
Test set: a set of examples used only to assess the performance of a fully-trained classifier In the SVM case, we would use the test to estimate the error rate, FP rate or TP rate after we have chosen the final model.

DATA FLOW DIAGRAMS

Data Flow Diagram can also be termed as bubble chart. It is a pictorial or graphical form, which can be applied to represent the input data to a system and multiple functions carried out on the data and the generated output by the system.

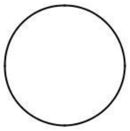
A graphical tool accustomed describe and analyze the instant of knowledge through a system manual or automatic together with the method, stores of knowledge, and delays within the system. The transformation of knowledge from input to output, through processes, is also delineate logically and severally of the physical elements related to the system. The DFD is also known as a data flow graph or a bubble chart.TheBasicNotation used to create a DFD's are as follows:

- **Dataflow:**

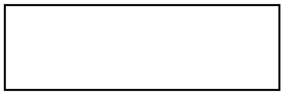


- **Process:**

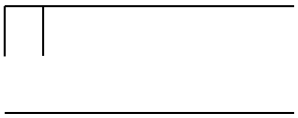
.



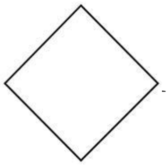
- **Source:**



- **Data Store:**



- **Rhombus**: decision



UML DIAGRAMS

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

This view represents the system from the users perspective. The analysis representation describes a usage scenario from the end-users perspective.

Structural Model view

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

USE CASE DIAGRAM :

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

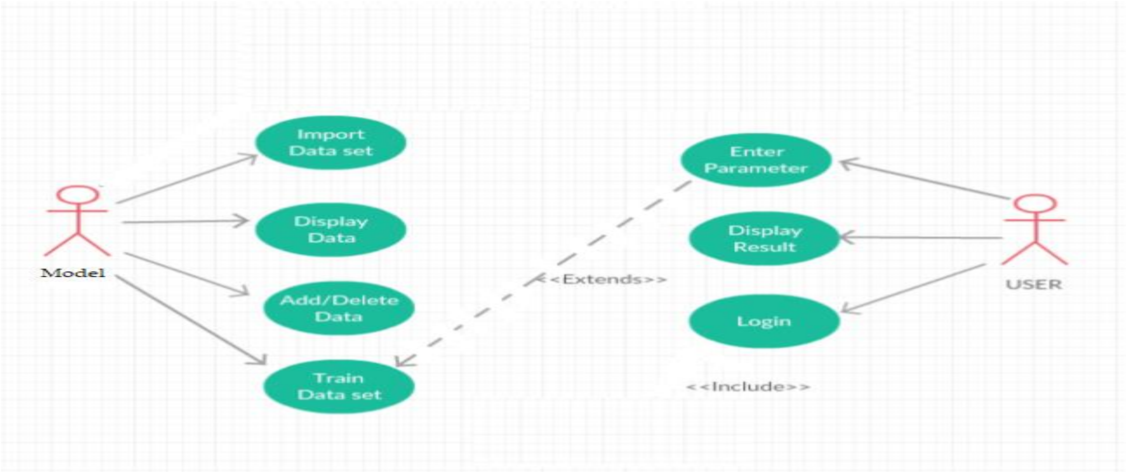


Figure.4.3.2: Use Case Diagram

CLASS DIAGRAM :

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. A class with three sections, in the diagram, classes is represented with boxes which contain three parts:

The upper part holds the name of the class

The middle part contains the attributes of the class

The bottom part gives the methods or operations the class can take or undertake.

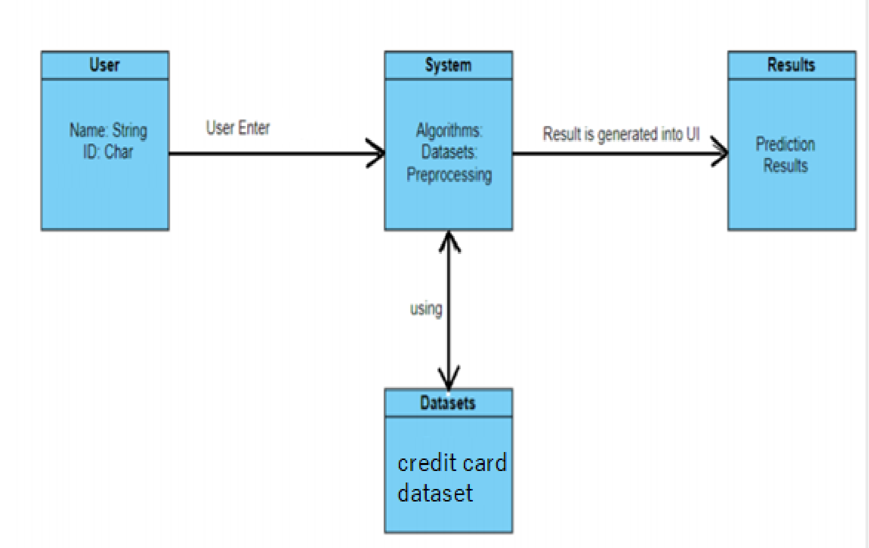


Figure.4.3.2:Class Diagram

SEQUENCEDIAGRAM:

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

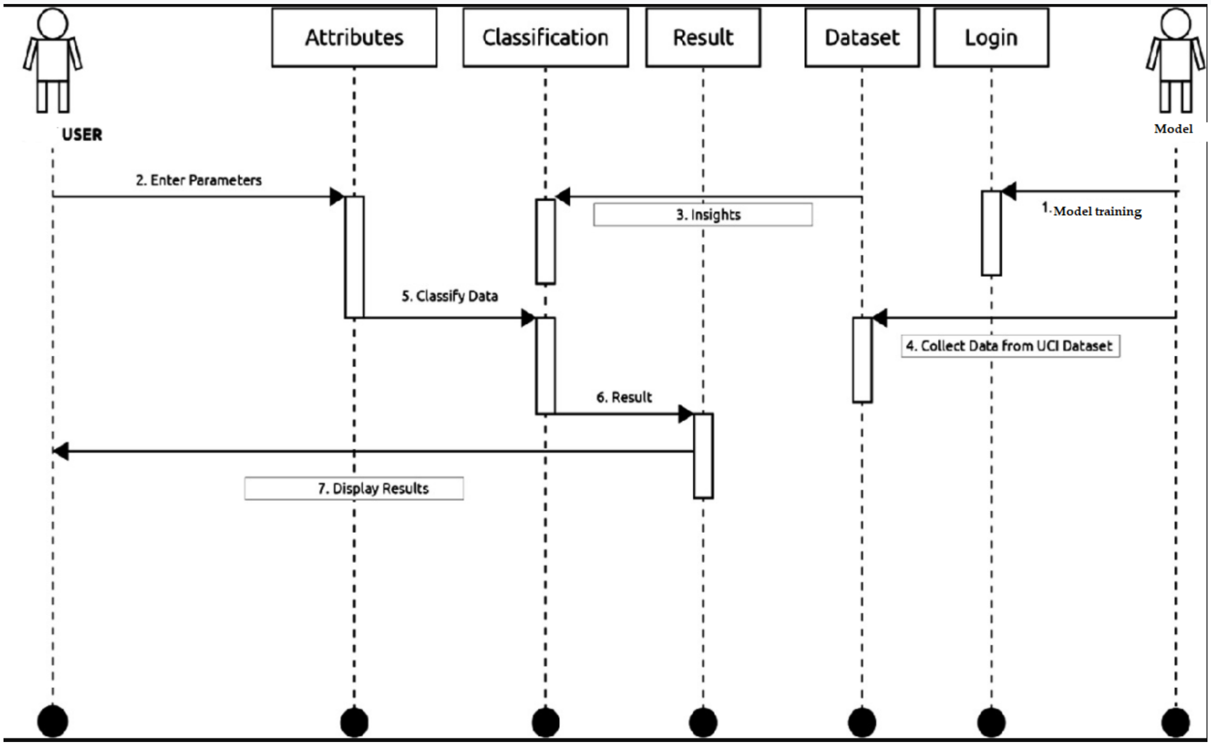


Figure.4.3.3: Sequence diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

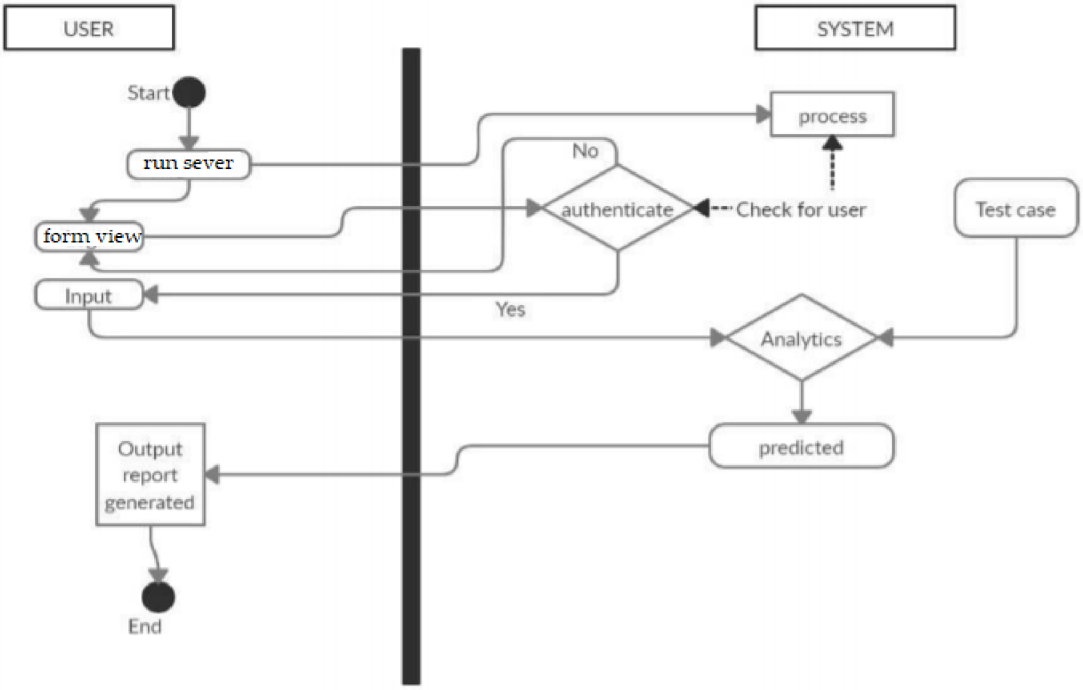


Figure.4.3.4 : Activity Diagram

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

EXISTING SYSTEM:

Existing system is amanual & time taking process.

PROPOSED SYSTEM:

The stored image file is completely secured, as the file is being encrypted not by just using one but three encryption algorithm which is AES, DES and RC6.

Advantages:

- .The key is also safe as it embeds the key in image using LSB.
- The system is very secure and robust in nature.
- Data is kept secured on cloud server which avoids unauthorized access.

Disadvantages:

- Requires an active internet connection to connect with cloud server.

Application:

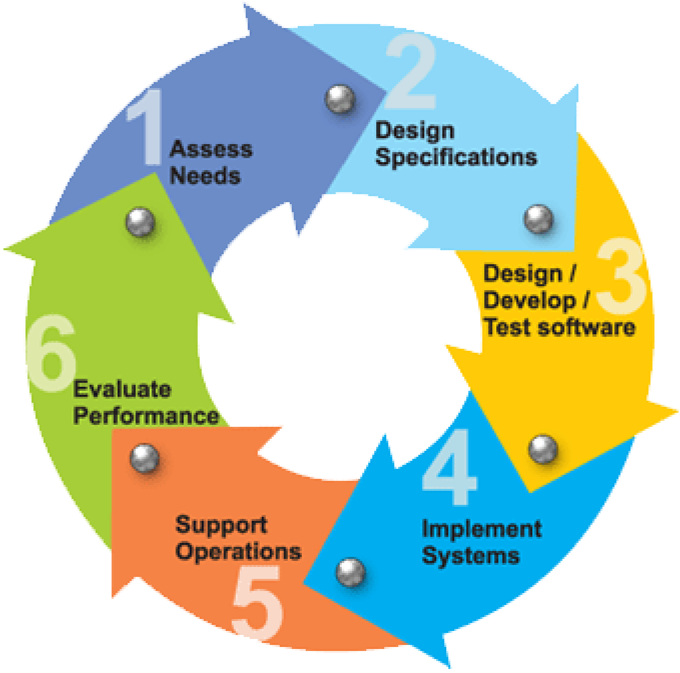
Data security is a major priority. This system can be implemented into banking and corporate sectors to securely transfer confidential data.

IMPLEMENTATION

INTRODUCTION

Software Development Life Cycle:

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development.



Requirements:

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.

Design:

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

Implementation:

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

Testing:

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

So in a nutshell, that is a very basic overview of the general software development life cycle model. Now let’s delve into some of the traditional and widely used variations.

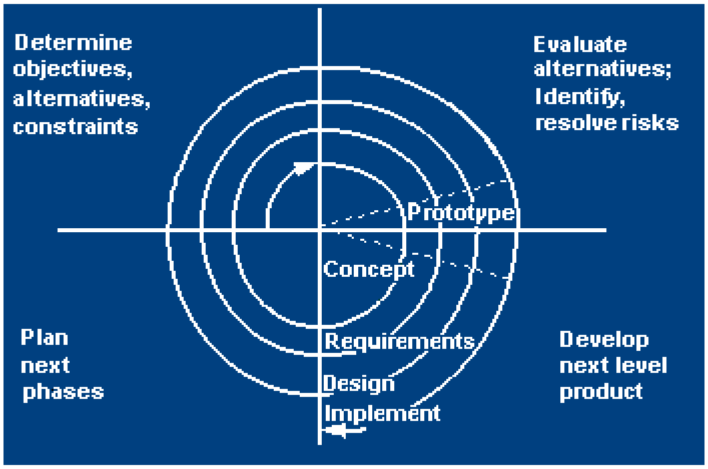
SDLC METHDOLOGIES:

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The following diagram shows how a spiral model acts like:


Determine objectives, alternatives, constraints
Evaluate alternatives; Identify, resolve risks
Prototype
Concept
Requirements
Design
Implement
Plan next phases
Develop next level product

The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible. This usually involves interviewing a number of usersrepresenting all the external or internal users and other aspects of the existing system.

- A preliminary design is created for the new system.

- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

- A second prototype is evolved by a fourfold procedure:

    1. Evaluating the first prototype in terms of its strengths, weakness, and risks.

    2. Defining the requirements of the second prototype.

    3. Planning a designing the second prototype.

4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

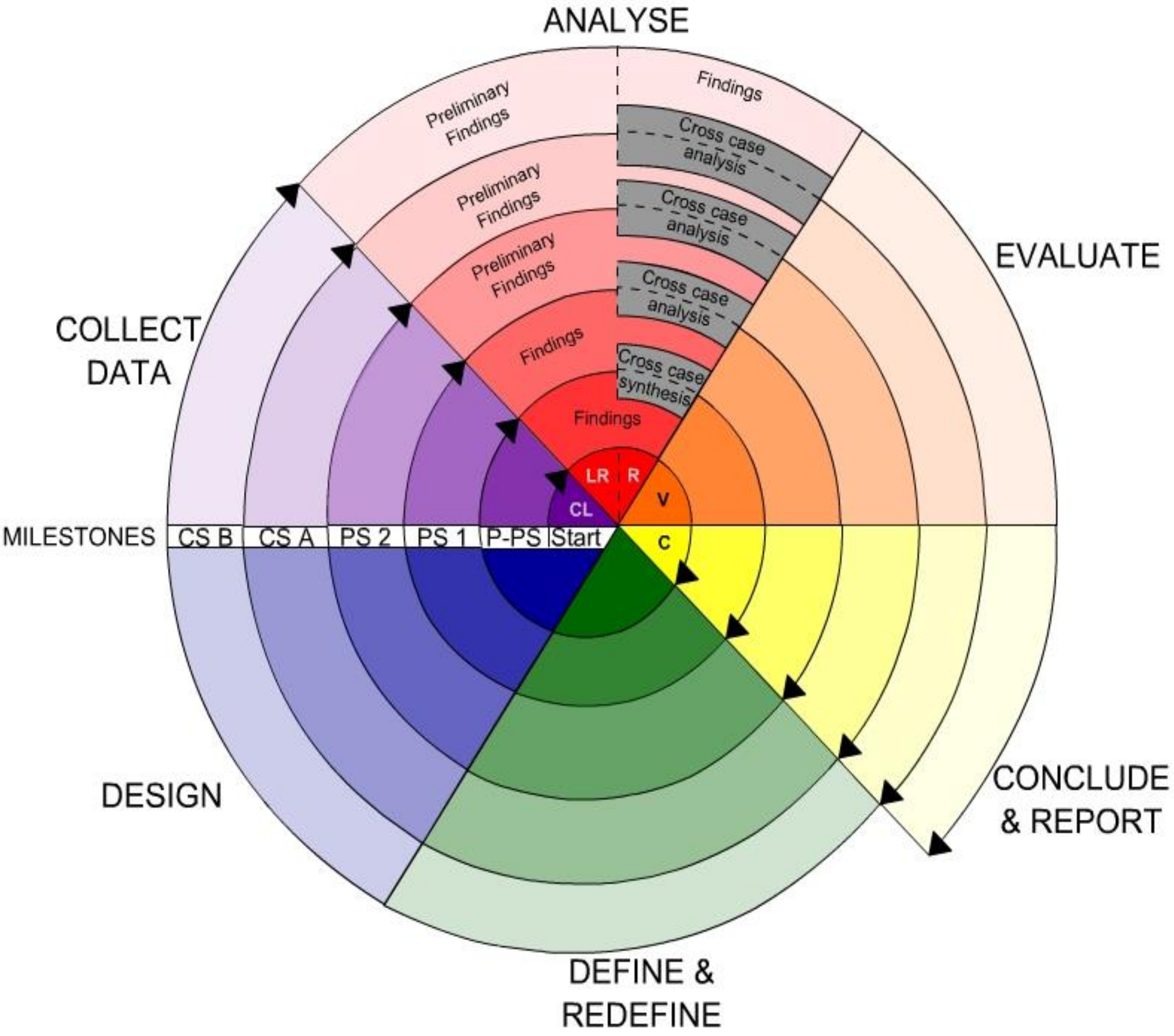
STUDY OF THE SYSTEM

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows

1. Administrative User Interface Design
2. The Operational and Generic User Interface Design

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.

The operational and generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.


ANALYSE
EVALUATE
COLLECT DATA
MILESTONES
CS B
CS A
PS 2
PS 1
P-PS
Start
DESIGN
DEFINE & REDEFINE
CONCLUDE & REPORT
Preliminary Findings
Preliminary Findings
Preliminary Findings
Findings
Findings
Findings
Cross case analysis
Cross case analysis
Cross case analysis
Cross case synthesis
LR
R
V
CL
C

INPUT AND OUTPUT

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department’s communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUTMEDIA:

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.

- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

OUTPUT DEFINITION:

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

Fundamental Concepts on (Domain)

Characteristics Of The Support Vector Machine (Svm) :

Support Vector Machine (Svm) Is A Popular Machine Learning Method For Classification, Regression, And Other Learning Tasks. The Basic Idea Of Svm Method Is: The Definition Of The Optimal Linear Hyper Plane, And The Search Algorithm For Optimal Linear Hyper Plane By Solving A Convex Programming Problem. Then Based On Mercer Nuclear Expansion Theorem, Through A Nonlinear Mapping , The Sample Space Is Mapped To A High-Dimensional And Even Infinite Dimensional Feature Space (Hilbert Space), So That In The Feature Space Can Be Applied To Solve The Linear Learning Machine Method, The Sample Space, The Highly Nonlinear Classification And Regression Problem. This Principle Is Based On The Fact That: The Error Rate Of Svm Depends On The Sum Of Training Error And An Item Relies On Vc Dimension. In The Case Of Classification, Support Vector Machines Make The Previous Value A Value Of Zero, And Make The Second Minimum. So Svm Has Good Generalization Performance In Pattern Classification.

Support Vector Machines Selection:

If you want to apply SVM to specific problems, you need to select a kernel function. Although theoretically those functions satisfying the Mercer condition can be selected as the kernel function, classification performance is completely different when using different kernel function. Thus, for a specific problem, selecting a specific function is very important. And even a certain type of kernel function is selected; we need to choose the corresponding parameters, such as the polynomial order and the width of the radial function. Quadratic programming parameters such as C can also affect the classifier’s generalization ability. Model selection includes the kernel function parameter selection, category selection and quadratic programming parameter selection. Although the research on model selection is not a lot, it is paid more attention by researchers as an important topic.

The Grid Search Method Is Often Used To Determine The Best Model. Its Basic Idea Is To Remove The Value Of A Number Of Model Parameters Respectively, And Combine Them Into A Number Of Combinations, And Then Train The Svm, Estimate The Accuracy Of Its Learning, And Finally Find A Combination Of The Most Learning Accuracy As The Optimal Combination Of Parameters. This Topic Uses The Radial Basis Kernel Function And The Grid Method To Determine The Optimal Parameters (C,). The Following Is The Specific Process.

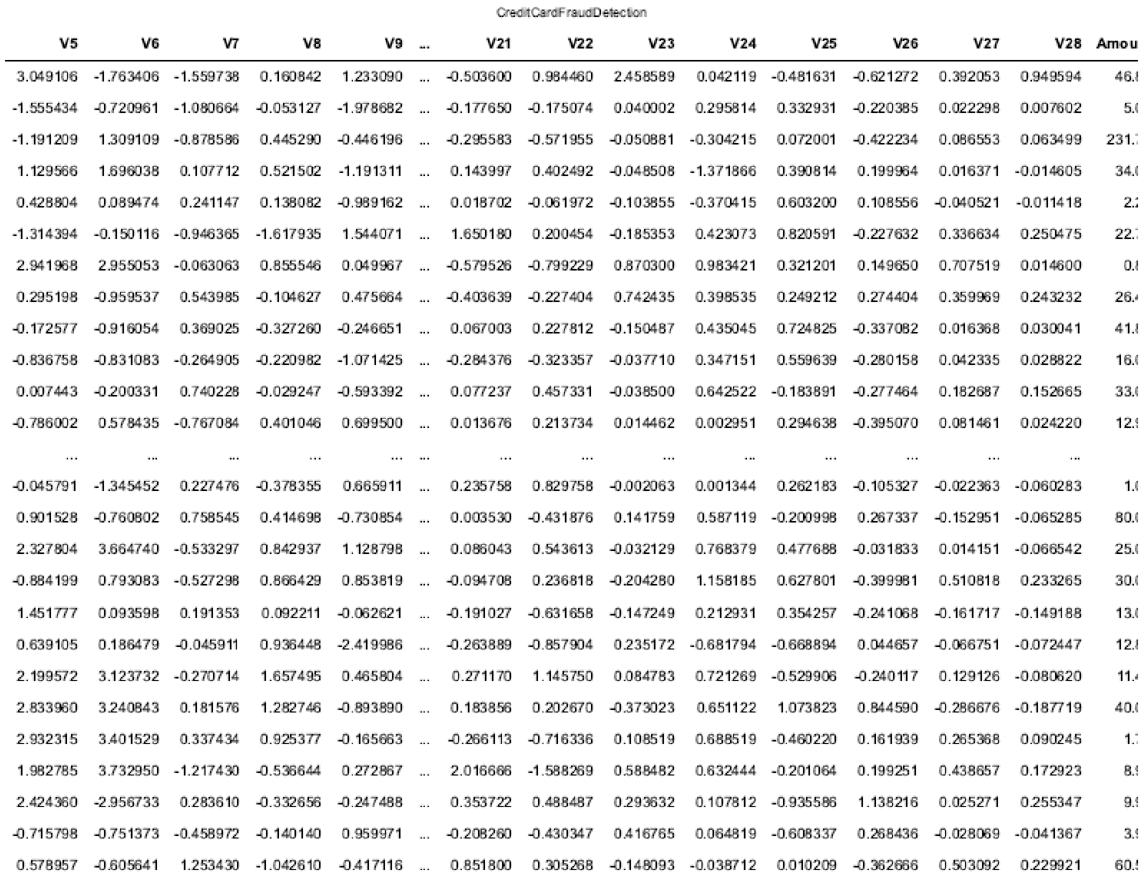
Set the range and step size of (C, ————————————). (C, ————————————) generally take the value of the index type C range from 100 to 1500 step 100; value of ——————— is 0.5 to 8; the step size is 0.5. Then a two-dimensional grid is constituted in the coordinate system.

• The value of the grid in each group (C, ————————————) will be in accordance with the following method to calculate the accuracy of the prediction. Sample data is divided into training and test set. Test set is used test SVM classifier fostered by the training set. Support Vector Machine classification in the latter classification accuracy is treated as the actual performance of support vector machine on the unknown data.

• Finally, the accuracy of each group (C, ————————————) values is depicted by the contour lines, getting a contour map on the accuracy. According to SRM theory, the highest point in the figure is the most advantages and determines the optimal (C, ————————————) values.

Code Snippets with Explanation:

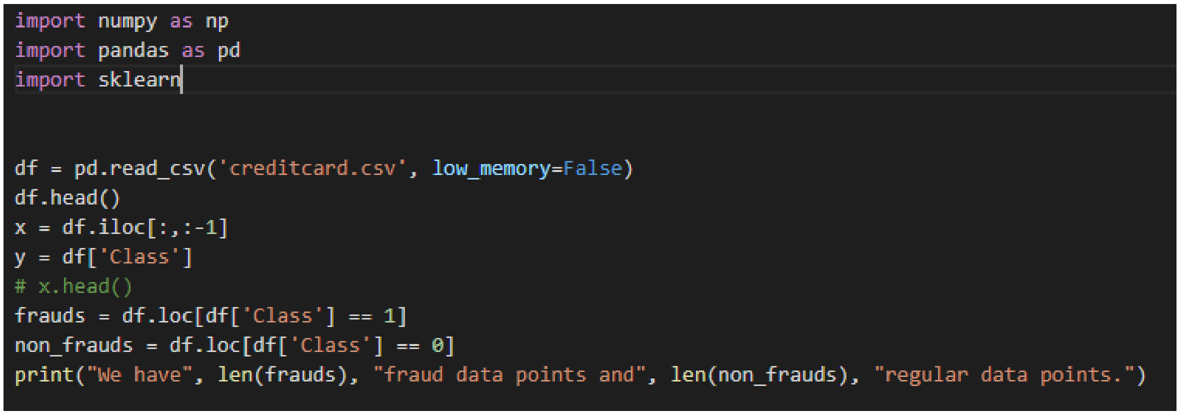
Dataset features:



CreditCardFraudDetection

V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amou
3.049106	-1.763406	-1.559738	0.160842	1.233090	...	-0.503600	0.984460	2.458589	0.042119	-0.481631	-0.621272	0.392053	0.949594	46.
-1.555434	-0.720961	-1.080664	-0.053127	-1.978682	...	-0.177650	-0.175074	0.040002	0.295814	0.332931	-0.220385	0.022298	0.007602	5.
-1.191209	1.309109	-0.878586	0.445290	-0.446196	...	-0.295583	-0.571955	-0.050881	-0.304215	0.072001	-0.422234	0.086553	0.063499	231.
1.129566	1.696038	0.107712	0.521502	-1.191311	...	0.143997	0.402492	-0.048508	-1.371866	0.390814	0.199964	0.016371	-0.014605	34.
0.428804	0.089474	0.241147	0.138082	-0.989162	...	0.018702	-0.061972	-0.103855	-0.370415	0.603200	0.108556	-0.040521	-0.011418	2.
-1.314394	-0.150116	-0.946365	-1.617935	1.544071	...	1.650180	0.200454	-0.185353	0.423073	0.820591	-0.227632	0.336634	0.250475	22.
2.941968	2.955053	-0.063063	0.855546	0.049967	...	-0.579526	-0.799229	0.870300	0.983421	0.321201	0.149650	0.707519	0.014600	0.
0.295198	-0.959537	0.543985	-0.104627	0.475664	...	-0.403639	-0.227404	0.742435	0.398535	0.249212	0.274404	0.359969	0.243232	26.
-0.172577	-0.916054	0.369025	-0.327260	-0.246651	...	0.067003	0.227812	-0.150487	0.435045	0.724825	-0.337082	0.016368	0.030041	41.
-0.836758	-0.831083	-0.264905	-0.220982	-1.071425	...	-0.284376	-0.323357	-0.037710	0.347151	0.559639	-0.280158	0.042335	0.028822	16.
0.007443	-0.200331	0.740228	-0.029247	-0.593392	...	0.077237	0.457331	-0.038500	0.642522	-0.183891	-0.277464	0.182687	0.152665	33.
-0.786002	0.578435	-0.767084	0.401046	0.699500	...	0.013676	0.213734	0.014462	0.002951	0.294638	-0.395070	0.081461	0.024220	12.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
-0.045791	-1.345452	0.227476	-0.378355	0.665911	...	0.235758	0.829758	-0.002063	0.001344	0.262183	-0.105327	-0.022363	-0.060283	1.
0.901528	-0.760802	0.758545	0.414698	-0.730854	...	0.003530	-0.431876	0.141759	0.587119	-0.200998	0.267337	-0.152951	-0.065285	80.
2.327804	3.664740	-0.533297	0.842937	1.128798	...	0.086043	0.543613	-0.032129	0.768379	0.477688	-0.031833	0.014151	-0.066542	25.
-0.884199	0.793083	-0.527298	0.866429	0.853819	...	-0.094708	0.236818	-0.204280	1.158185	0.627801	-0.399981	0.510818	0.233265	30.
1.451777	0.093598	0.191353	0.092211	-0.062621	...	-0.191027	-0.631658	-0.147249	0.212931	0.354257	-0.241068	-0.161717	-0.149188	13.
0.639105	0.186479	-0.045911	0.936448	-2.419986	...	-0.263889	-0.857904	0.235172	-0.681794	-0.668894	0.044657	-0.066751	-0.072447	12.
2.199572	3.123732	-0.270714	1.657495	0.465804	...	0.271170	1.145750	0.084783	0.721269	-0.529906	-0.240117	0.129126	-0.080620	11.
2.833960	3.240843	0.181576	1.282746	-0.893890	...	0.183856	0.202670	-0.373023	0.651122	1.073823	0.844590	-0.286676	-0.187719	40.
2.932315	3.401529	0.337434	0.925377	-0.165663	...	-0.266113	-0.716336	0.108519	0.688519	-0.460220	0.161939	0.265368	0.090245	1.
1.982785	3.732950	-1.217430	-0.536644	0.272867	...	2.016666	-1.588269	0.588482	0.632444	-0.201064	0.199251	0.438657	0.172923	8.
2.424360	-2.956733	0.283610	-0.332656	-0.247488	...	0.353722	0.488487	0.293632	0.107812	-0.935586	1.138216	0.025271	0.255347	9.
-0.715798	-0.751373	-0.458972	-0.140140	0.959971	...	-0.208260	-0.430347	0.416765	0.064819	-0.608337	0.268436	-0.028069	-0.041367	3.
0.578957	-0.605641	1.253430	-1.042610	-0.417116	...	0.851800	0.305268	-0.148093	-0.038712	0.010209	-0.362666	0.503092	0.229921	60.

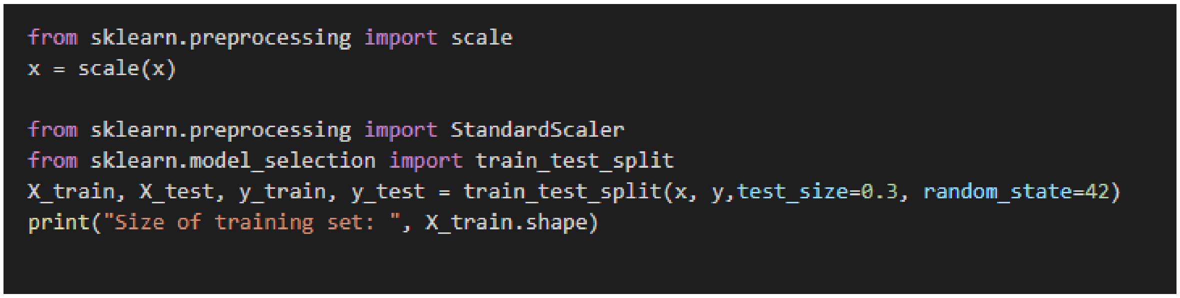
Loading Dataset and collecting features and labels in x, y variables.



```
import numpy as np
import pandas as pd
import sklearn


df = pd.read_csv('creditcard.csv', low_memory=False)
df.head()
x = df.iloc[:,:-1]
y = df['Class']
# x.head()
frauds = df.loc[df['Class'] == 1]
non_frauds = df.loc[df['Class'] == 0]
print("We have", len(frauds), "fraud data points and", len(non_frauds), "regular data points.")
```

Preprocessing Stage:

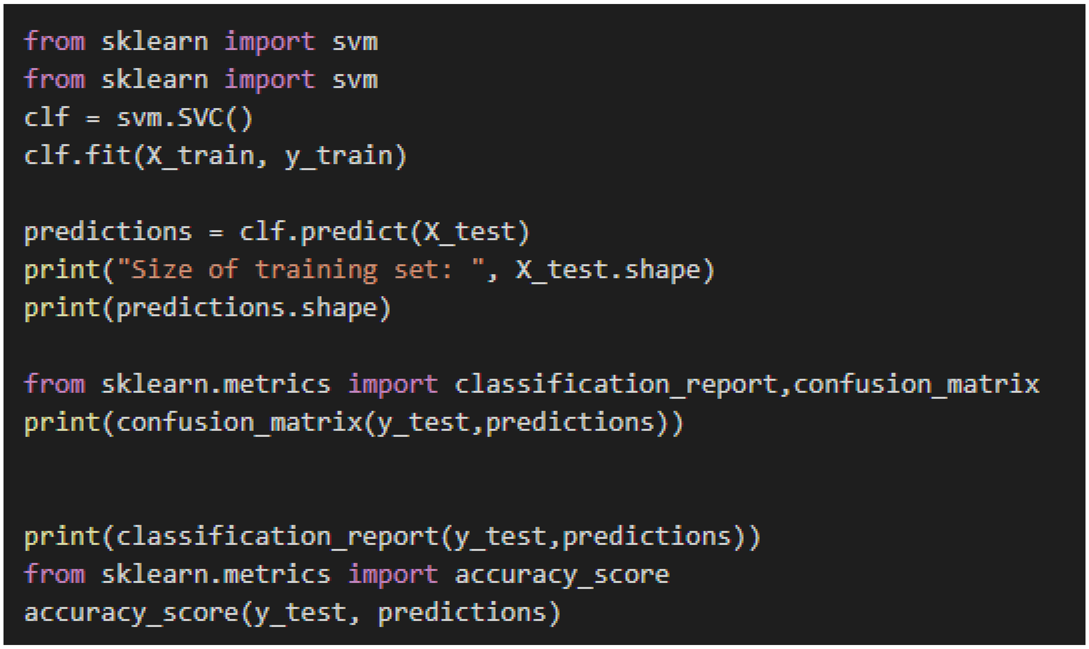


```
from sklearn.preprocessing import scale
x = scale(x)

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y,test_size=0.3, random_state=42)
print("Size of training set: ", X_train.shape)
```

Dividing data in to train and test sets as shown in code using train_test_split functions.

SVM Algorithm Loading and accuracy calcuation:



```
from sklearn import svm
from sklearn import svm
clf = svm.SVC()
clf.fit(X_train, y_train)

predictions = clf.predict(X_test)
print("Size of training set: ", X_test.shape)
print(predictions.shape)

from sklearn.metrics import classification_report,confusion_matrix
print(confusion_matrix(y_test,predictions))


print(classification_report(y_test,predictions))
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predictions)
```

By initializing svm algorithm train data is given as input to fit function for svm algorithm and then predict function is called with test set which has half of train records features based on output prediction values will be labels of test set and then accuracy is calculated using accuracy score function by comparing with existing test labels in dataset.

8. TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

8.1 SYSTEM TESTING

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

8.2 MODULE TESTING

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

8.3 INTEGRATION TESTING

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

8.4 ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

8.5 TEST CASES:

Test Case Id	**Test Case Name**	**Test Case Desc.**	**Test Steps**			**Test Case Status**	**Test Priority**
			Step	**Expected**	**Actual**		
01	Upload credit card dataset	Verify if dataset has data	If dataset is not uploaded	It cannot display dataset reading process completed	It can display dataset reading process completed	High	High
02	Extract features and labels	Verify if features and labels are stored	If x and y has no values	x cannot print features values y cannot print label values	It can display features in x and labels in y	low	High
03	Preprocessing	Whether preprocessing on the dataset applied or not	If not applied	It cannot display the necessary data for further process	It can display the necessary data for further process	Medium	High

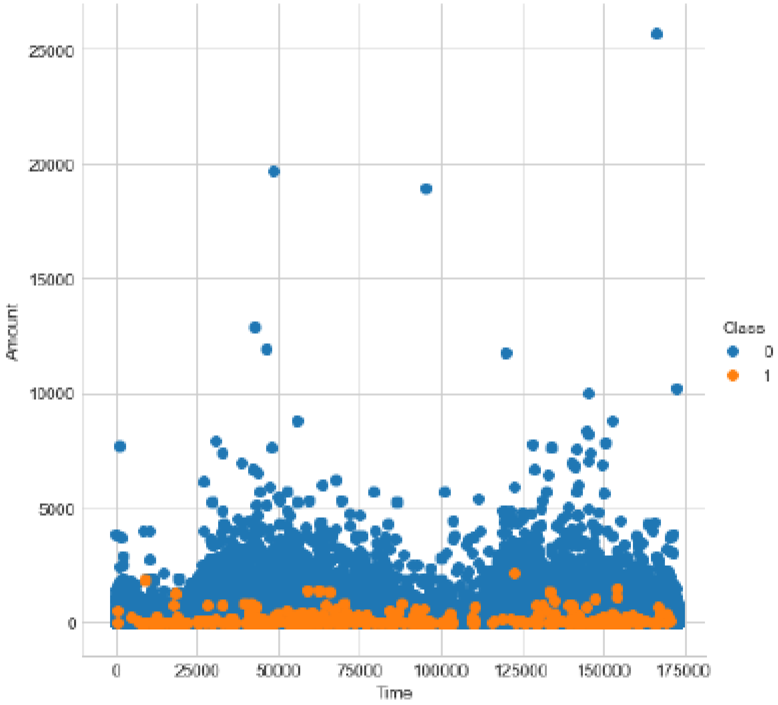
04	Prediction SVM	Whether Prediction algorithm applied on the data or not	If not applied	Random tree is not generated	Random tree is generated	High	High
05	Accuracy calcuation	Whether predicted data is displayed or not	If not displayed	It cannot view prediction containing patient data	It can view prediction containing patient data	High	High
06	Noisy Records Chart	Whether the graph is displayed or not	If graph is not displayed	It does not show the variations in between clean and noisy records	It shows the variations in between clean and noisy records	Low	Mediu m

TABLE 8.5.1 TESTCASES

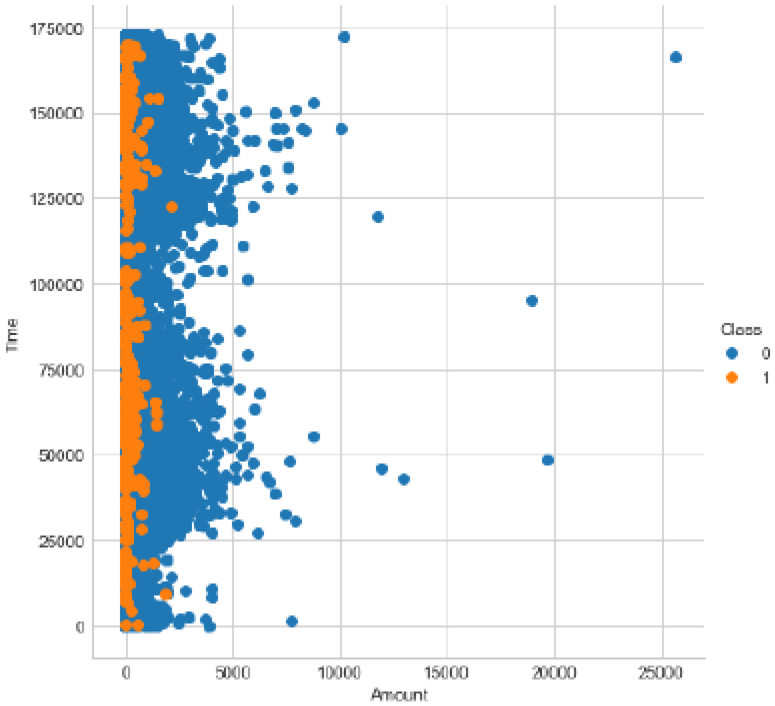
RESULTS

Data Analysis:

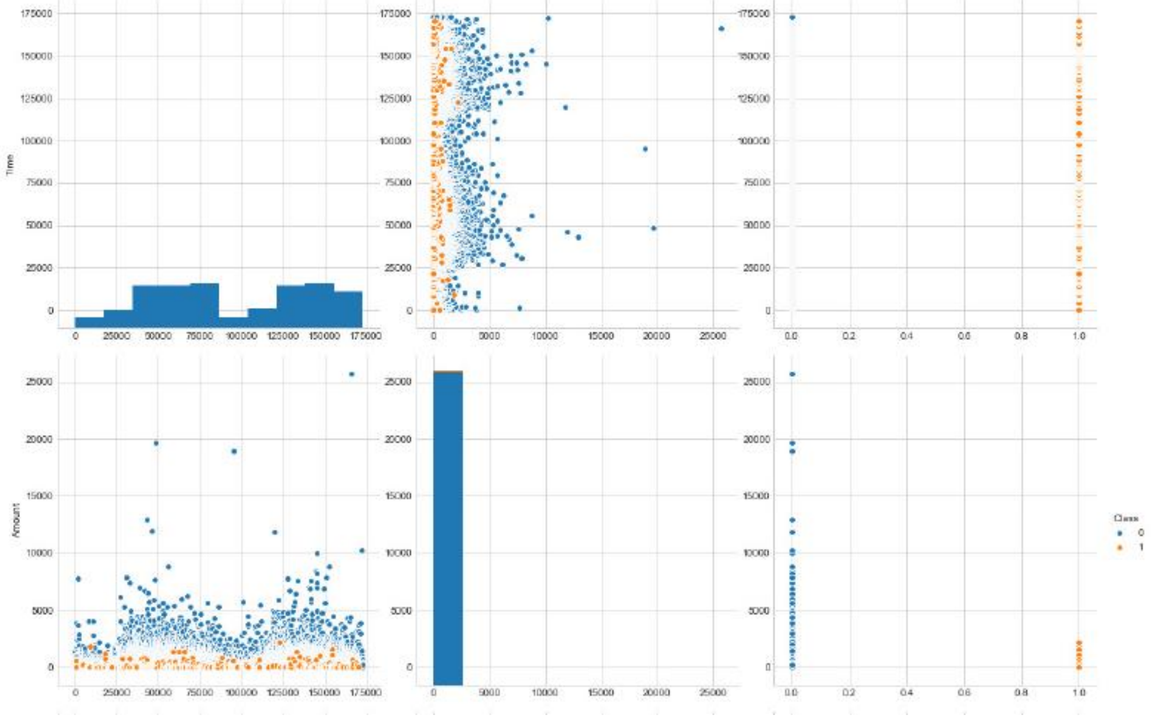
Total dataset view:



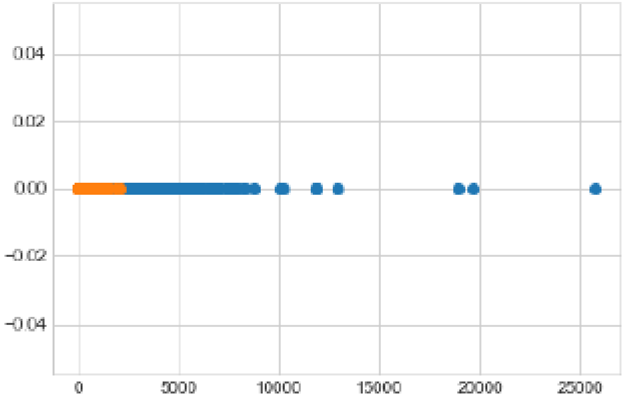
2D Scatter Plot with time and Amount:



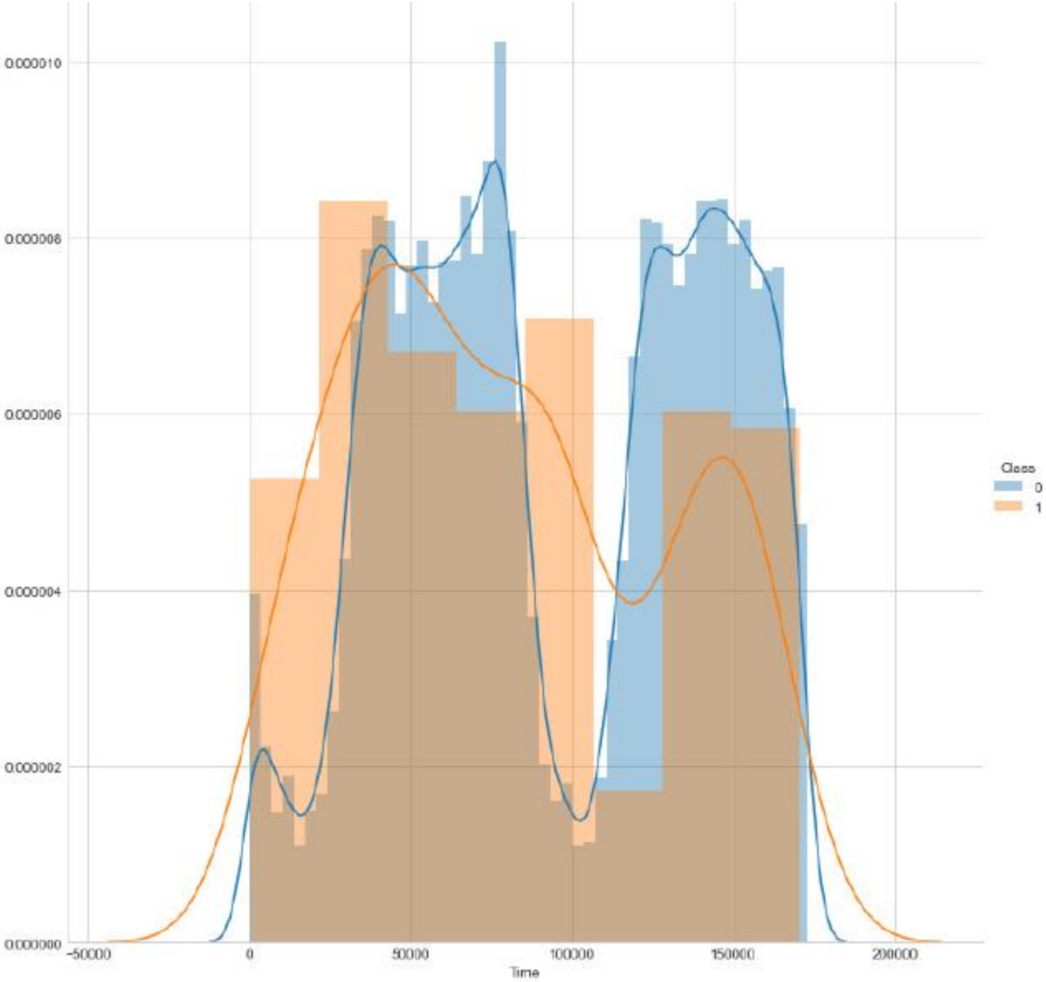
3D scatter plot for transactions above 2500 and below 2500 Amount:



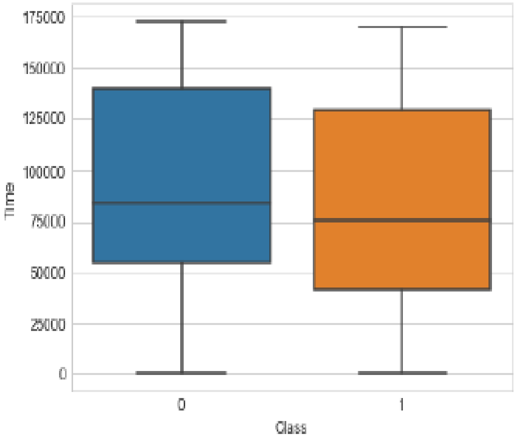
Fraud Transactions below 25000



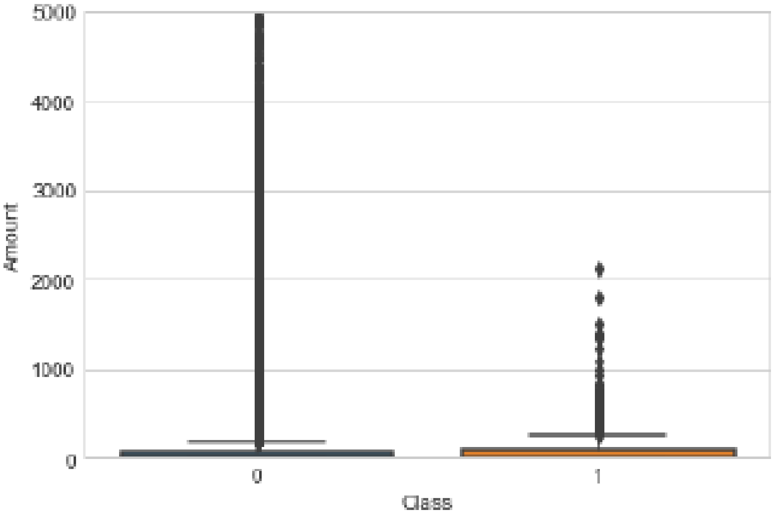
Comparing Genuine and Fraud Transactions :



Total Fraud and Genuine Transactions:



No fraud Transactions about 30000:



CONCLUSION

This work examined the performance of advanced data mining techniques support vector machines, for credit card fraud detection. For performance assessment, we use a test dataset with much lower fraud rate (0.5%) than in the training datasets with different levels of under sampling. This helps provide an indication of performance that may be expected when models are applied for fraud detection where the proportion of fraudulent transactions are typically low.SVM predicts 94.3% customers correctly; only 6.7% true bad customers are predicted as good customers; and 13.3% true good customers are predicted as bad ones. To compare single tree data mining method with ensemble methods, considering the two wrongly prediction situations the same bad, SVM, bagging, boosting, and random forest are also applied into this dataset. All methods tell us a customer's checking account existing status and duration time are important variables to predict his or her credit risk. Without exception, ensemble methods have lower misclassification rates than the single tree method SVM where bagging shows the best predicting result that 94.3.7% customers in the test sample are predictedcorrectly.

Future Enhancements:

Future research can explore possibilities for creating ingenious derived attributes to help classify more transactions more accurately. We created derived attributes based on past research, but future work can usefully undertake a broader study of attributes best suited for fraud modeling, including the issue of transaction aggregation. Another interesting issue for investigation is how the fraudulent behavior of a card with multiple fraudulent transactions is different from a card with few fraudulent transactions. As mentioned above, a limitation in our data was the non- availability of exact time stamp data beyond the date of credit card transactions. Future study may focus on the difference in sequence of fraudulent and legitimate transactions before a credit card is withdrawn. Future research may also examine differences in fraudulent behavior among different types of fraud, say the difference in behavior between stolen and counterfeit cards. Alternative means for dividing the data into training and test remains another issue for investigation. The random sampling of data into training and test as used in this study assumes that fraud patterns will remain essentially same over the anticipated time period of application of such patterns. Given the increasingly sophisticated mechanisms being applied by fraudsters and the potential for their varying such mechanisms over time to escape detection, such assumptions of stable patterns over time may not hold. Consideration of data drift issues can then become important. To better match how developed models may be used in real application, training and test data can be set up such that trained models are tested for their predictive ability in subsequent time periods. With availability of data covering a longer time period, it will be useful to examine the extent of concept drift and whether fraud pattern effect overtime.

REFERENCES

- Ming Xiao, The anti-fraud reserch for credit card online payment, Nankai University, May 2010.

- [2] Pai, Ping-Feng, Chih-Shen Lin, A hybrid ARIMA and support vector machines model in stock price forecasting, Omega: International Journal of Management Science, 2005, 33(6): 497-505.

- [3] Linhui Li, Intrusion Detection Based on Feature Selection, Zhongnan forestry university of science and technology, 2009.

- [4] Ling Yang, Tongue color pattern recognition system, Nankai University, 2008.

- [5] C. Chiu, C. Tsai: A Web Services-Based Collaborative Scheme for Credit Card Fraud Detection[C].Proceedings of 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service,2004:177-181.

- [6] Daqin Wei, Detection of risk of credit card transactions based on data mining, Chengdu: Sichuan Normal University, 2007.

- Delamaire,L., Abdou,H., and Pointon,J.,(2009) "Credit card fraud and detection techniques: a review." Banks and Bank systems ,4(2),pp. 57-68.

- Phua,C., Lee,V., Smith,K., and Gayler,R.,(2010) "A comprehensive survey of data mining-based fraud detection research." arXiv preprint arXiv:1009.6119.
- Raj, S., and A. Annie Portia(2011). "Analysis on credit card fraud detection methods." In International Conference On Computer, Communication and Electrical Technology (ICCCET),IEEE, pp.152-156.
- Sahin, Y., and Duman,E., (2011)"Detecting credit card fraud by decision trees and support vector machines," In International Multi conference of Engineers and computer scientists,(1).
- Ganji,V.R., and Mannem,S.N.P.,(2012) "Credit card fraud detection using anti-k nearest neighbor algorithm." International Journal on Computer Science and Engineering ,4(6),pp.1035.
- Chaudhary, K.,Yadav,J., and Malik.B.,(2012) "A review of fraud detection techniques: Credit card," International Journal of Computer Applications ,(0975–8887).
- Dhok, Shailesh S., and Bamnote,G.R.,(2012) "Credit card fraud detection using hidden markov model." International Journal of Advanced Research in Computer Science ,2(4).
- Richhariya,P.,Singh,P.K.,(2012),"A Survey on Financial Fraud Detection Methodologies."International Journal of Computer Applications, 45,pp.22.
- Singh, A., and Narayan,D.,(2012) "A survey on hidden markov model for credit card fraud detection." International Journal of Engineering and Advanced Technology (IJEAT),pp.49-52.
- Tripathi,K.K.,andPavaskar,M.A.,(2012), "Survey on credit card fraud detection methods." International Journal of Emerging Technology and Advanced Engineering,2(11),pp.721-726.
- Akhilomen, J.,(2013) "Data mining application for cyber credit-card fraud detection system," In Advances in Data Mining. Applications and Theoretical pp.218-228.

BIBLIOGRAPHY

python Technologies

python Complete Reference

*python*Script Programming by Yehuda Shiran

Mastering *python*Security

*python*Professional by Shadab siddiqui

*python*server pages by Larne Pekowsley

*python*Server pages by Nick Todd

HTML

HTML Black Book by Holzner

sqllite