CREDIT CARD FRAUD DETECTION USING RANDOM FOREST ALGORITHM

SUBMITTED TO

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA

In the partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

Submitted by

P. SRAVANA GOWRI	**18NG1A0533**
V. YOGNA SRI	**18NG1A0552**
P. D V SATHVIKA REDDY	**18NG1A0537**
Y. N S SAIRAM SATHVIK	**18NG1A0557**

Under the Esteemed Guidance of

Mr. CH. PHANI KUMAR, M.Tech

Assistant Professor

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Approved by AICTE and JNTUK, Kakinada)

(ON NH 16, TELAPROLU, NEAR GANNAVARAM - 521109)

2018-2022

AUTONOMOUS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this project entitled **"CREDIT CARD FRAUD DETECTION USING RANDOM FOREST ALGORITHM"** is the Bonafide work of **P. Sravana Gowri (18NG1A0533), V. Yogna Sri (18NG1A0552), P. D V Sathvika Reddy(18NG1A0537), Y. N S Sairam Sathvik (18NG1A0557)** who carried out the work under my supervision, and submitted in partial fulfilment of the requirements for the award of the degree in Bachelor of Technology in Computer Science & Engineering, during the academic year 2018-22.

Project Guide
Mr. CH. PHANI KUMAR

Head of the Department
Dr. S M ROY CHOUDRI

Signature of External Examiner

DECLARATION

We hereby declare that the project entitled **"CREDIT CARD FRAUD DETECTION USING RANDOM FOREST ALGORITHM"** is the work done by us during the academic year 2018-2022 and is submitted in partial fulfilment of the requirements for the award of degree of **Bachelor of technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA**.

BY

P. Sravana Gowri	**(18NG1A0533)**
V. Yogna Sri	**(18NG1A0552)**
P. D V Sathvika Reddy	**(18NG1A0537)**
Y. N S Sairam Sathvik	**(18NG1A0557)**

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to our Honorable Chairman **SRI.S. RAMABRAHMAM** for the guidance and advice which is given and for providing sufficient resources.

We are extremely thankful to **Dr. K RAJASEKHARA RAO**, Director of USHA RAMA COLLEGE OF ENGINEERING AND TECHNOLOGY, TELAPROLU for giving a golden opportunity to our education and project work.

We wish to avail this opportunity to express to thank **Dr. G V K S V PRASAD**, Principal, URCE for his continuous support and giving valuable suggestions during the entire period of the project work.

We take this opportunity to express our gratitude to **Dr. S M ROY CHOUDRI**, Head of the Department and also our guide **Mr. CH. PHANI KUMAR**, Assistant Professor in **Computer Science and Engineering** for his valuable support and motivation at each and every point in successful completion of the project.

We also place our floral gratitude to all other teaching staff and lab technicians for their constant support and advice throughout the project.

BY

P. Sravana Gowri (18NG1A0533)

V. Yogna Sri (18NG1A0552)

P. D V Sathvika Reddy (18NG1A0537)

Y. N S Sairam Sathvik (18NG1A0557)

CREDIT CARD FRAUD DETECTION USING RANDOM FOREST ALGORITHM

ABSTRACT

ABSTRACT

Credit card fraud is increasing day by day. Credit card fraud can be done in both online and offline transactions. In offline transactions Physical cards are required while in online transactions the virtual cards are required for doing illegal or fraud activities. Thus, these fraud activities in credit card may lead to many fraud transactions without the knowledge of the actual users. The fraudsters are looking for sensitive information such as credit card number, bank account and other user details in order to perform transactions.

There are many fraud transactions which cannot be easily identified by the user and also by the banking authority which leads to loss of sensitive data. There are various models which are used for detecting the fraud transactions based on the behavior of the transactions and these methods can be classified as two broad categories such as supervised learning and unsupervised learning algorithm. In existing system for finding the accuracy of the fraudulent activates they have used methods such as Cluster Analysis, Support Vector Machine, Naïve Bayer's Classification etc. The aim of this Project is to detect the accuracy of the fraudulent transactions by using Random Forest Algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER - 1

INTRODUCTION

1. INTRODUCTION

There are various fraudulent activities detection techniques has implemented in credit card transactions have been kept in researcher minds to methods to develop models based on artificial intelligence, data mining, fuzzy logic and machine learning. Credit card fraud detection is significantly difficult, but also popular problem to solve. In our proposed system we built the credit card fraud detection using Machine learning. With the advancement of machine learning techniques. Machine learning has been identified as a successful measure for fraud detection. A large amount of data is transferred during online transaction processes, resulting in a binary result: genuine or fraudulent. Within the sample fraudulent datasets, features are constructed. These are data points namely the age and value of the customer account, as well as the origin of the credit card. There are hundreds of features and each contributes, to varying extents, towards the fraud probability. Note, the level in which each feature contributes to the fraud score is generated by the artificial intelligence of the machine which is driven by the training set, but is not determined by a fraud analyst. So, in regards to the card fraud, if the use of cards to commit fraud is proven to be high, the fraud weighting of a transaction that uses a credit card will be equally so. However, if this were to shrink, the contribution level would parallel. Simply make, these models self-learn without explicit programming such as with manual review. Credit card fraud detection using Machine learning is done by deploying the classification and regression algorithms. We use supervised learning algorithm such as Random forest algorithm to classify the fraud card transaction in online or by offline. Random forest is advanced version of Decision tree. Random forest has better efficiency and accuracy than the other machine learning algorithms. Random forest aims to reduce the previously mentioned correlation issue by picking only a subsample of the feature space at each split. Essentially, it aims to make the trees de-correlated and prune the trees by fixing a stopping criterion for node splits, which I will be cover in more detail later.

1.1. LITERATURE SURVEY

Along with increasing credit card and growing trade volume in credit card fraud rises sharply. How to enhance the detection and bar of credit card fraud becomes the main target of risk management of banks. This paper proposes a credit card fraud detection model victimization outlier detection supported distance add consistent with the scarceness and unconventionality of fraud in credit card dealing information, applying outlier mining into credit card fraud detection. Experiments show that this model is feasible and accurate in detecting credit card fraud. With growing advancement within the electronic commerce field, fraud is spreading all over the world, causing major financial losses. In current scenario, Major cause of financial losses is credit card fraud. It not only affects trades person but also individual clients. Decision tree, Genetic algorithm, Meta learning strategy, neural network, HMM are the presented methods used to detect credit card frauds. In contemplate system for fraudulent detection, artificial intelligence concept of Support Vector Machine (SVM) & decision tree is being used to solve the problem. Thus, by implementation of this hybrid approach, financial losses can be reduced to greater extend. In this paper, we tend to proposing the SVM (Support Vector Machine) primarily based methodology with multiple kernel involvement that additionally includes many fields of user profile rather than solely of only spending profile. The simulation result shows improvement in TP (true positive), TN (true negative) rate, & also decreases the FP (false positive) & FN (false negative) rate. In this study, classification models supported on decision trees and Support

Vector Machines (SVM) are developed and applied on credit card fraud detection problems. This study is one of the first to compare the performance of SVM and decision tree methods in credit card fraud detection with a real data set . A new cost-sensitive decision tree approach which reduces the sum of misclassification costs while selecting the splitting attribute at each non-terminal node is advanced and the act of this approach is compared with the well-known ancient classification models on a true world credit card data set. This analysis is completely involved with master card application fraud detection by performing arts the method of asking security queries to the persons byzantine with the transactions and as well as by eliminating real time data faults.

1.1.1. MACHINE LEARNING

Tom Mitchell states machine learning as “A computer program is said to learn from experience and from some tasks and some performance on, as measured by, improves with experience”. Machine Learning is combination of correlations and relationships, most machine learning algorithms in existence are concerned with finding and/or exploiting relationship between datasets. Once Machine Learning Algorithms can pinpoint on certain correlations, the model can either use these relationships to predict future observations or generalize the data to reveal interesting patterns. In Machine Learning there are various types of algorithms such as Regression, Linear Regression, Logistic Regression, Naive Bayes Classifier, Bayes theorem, KNN (K-Nearest Neighbor Classifier), Decision Tress, Entropy, ID3, SVM (Support Vector Machines), K-means Algorithm, Random Forest and etc.,

The name machine learning was coined in 1959 by Arthur Samuel. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with datamining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning.

Within the field of data analytics, machine learning is a method used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and " hidden insights" through learning from historical relationships and trends in the data.

Machine learning implementations are classified into three major categories, depending on the nature of the learning “signal” or “response” available to a learning system which are as follows:

Supervised learning: When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning. This approach is indeed similar to human learning under the supervision of a teacher. The teacher provides good examples for the student to memorize, and the student then derives general rules from these specific examples.

Unsupervised learning: When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms. As a kind of learning, it resembles the methods humans use to figure out that certain objects or events are from the same class, such as by observing the degree of similarity between objects. Some recommendation systems that you find on the web in the form of marketing automation are based on this type of learning.

Reinforcement learning: When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others.

In this case, an application presents the algorithm with examples of specific situations, such as having the gamer stuck in a maze while avoiding an enemy. The application lets the algorithm know the outcome of actions it takes, and learning occurs while trying to avoid what it discovers to be dangerous and to pursue survival. You can have a look at how the company Google Deep Mind has created a reinforcement learning program that plays old Atari's video 3 games. When watching the video, notice how the program is initially clumsy and unskilled but steadily improves with training until it becomes a champion.

Semi-supervised learning:

Where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing. Supervised Learning the majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data. It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Types of Supervised Learning:

Classification: It is a Supervised Learning task where output is having defined labels (discrete value). For example, in above Figure A, Output – Purchased has defined labels i.e., 0 or 1; 1 means the customer will purchase and 0 means that customer won't purchase. The goal here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy. It can be either binary or multi class classification. In binary classification, model predicts either 0 or 1; yes or no but in case of multi class classification, model predicts more than one class. Example: Gmail classifies mails in more than one classes like social, promotions, updates, forum.

Regression: It is a Supervised Learning task where output is having continuous value. Example in above Figure B, Output – Wind Speed is not having any discrete value but is continuous in the particular range. The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.

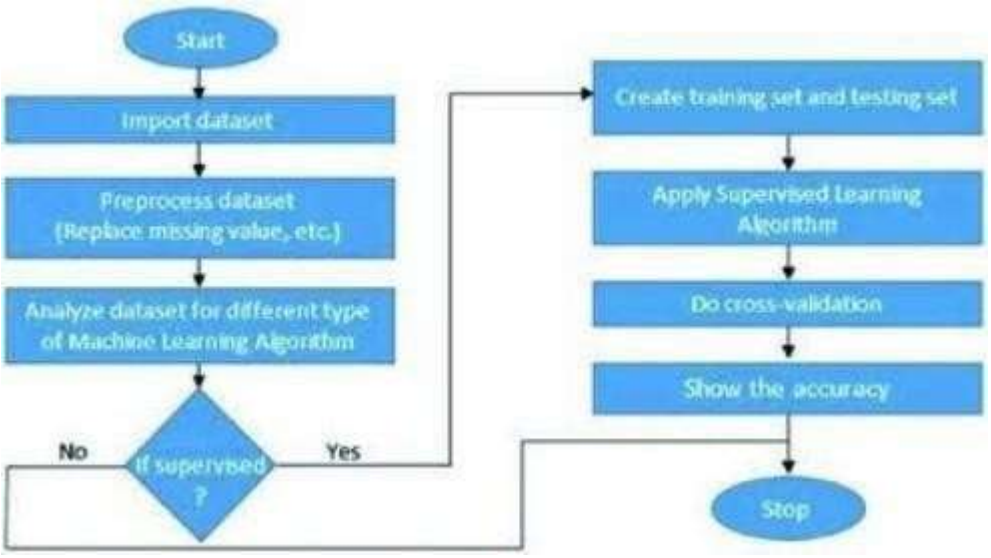


Fig 1.1.1. Flow Chart of Supervised Learning Algorithm

Classification:

Data mining is the process of extracting knowledge-able information from huge amounts of data. It is an integration of multiple disciplines such as statistics, machine learning, neural networks and pattern recognition. Data mining extracts biomedical and health care knowledge for clinical decision making and generates scientific hypotheses from large medical data.

Association rule mining and classification are two major techniques of data mining. Association rule mining is an unsupervised learning method for discovering interesting patterns and their association in large data bases.

Classification is a supervised learning method used to find class labels for unknown samples. Classification is the task of assigning an object's tone of special predefined categories. It is pervasive problem that encompasses many applications.

Classification is designed as the task of learning a target function F that maps each attribute set A to one of the predefined class labels C. The target function is also known as classification model.

A classification model is useful for mainly two purposes.

1) descriptive modelling.
2) Predictive modelling.

Classification is the process of recognizing, understanding, and grouping ideas and objects into pre-set categories or “sub-populations.” Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.

Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. One of the most common uses of classification is filtering emails into “spam” or “non-spam.”

In short, classification is a form of “pattern recognition,” with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.

Classification can be performed on structured or unstructured data. Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Few of the terminologies encountered in machine learning – classification:

Classifier: An algorithm that maps the input data to a specific category.

Classification model: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.

Feature: A feature is an individual measurable property of a phenomenon being observed. Binary

Classification: Classification task with two possible outcomes. E.g., Gender classification (Male / Female).

Multi-class classification: Classification with more than two classes. In multi class classification each sample is assigned to one and only one target label. E.g., An animal can be cat or dog but not both at the same time.

Multi-label classification: Classification task where each sample is mapped to a set of target labels (more than one class). E.g., A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms:

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drug's classification
- Facial key points detection
- Pedestrians' detection in an automotive car driving.

1.1.2. FEATURES OF MACHINE LEARNING

- It is nothing but automating the Automation.
- Getting computers to program themselves.
- Writing Software is bottleneck.
- Machine leaning models involves machines learning from data without the help of humans or any kind of human intervention.
- Machine Learning is the science of making of making the computers learn and act like humans by feeding data and information without being explicitly programmed.
- Machine Learning is totally different from traditionally programming, here data and output is given to the computer and in return It gives us the program which provides solution to the various problems. Below is the figure.

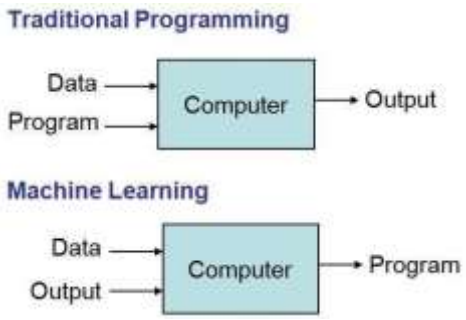
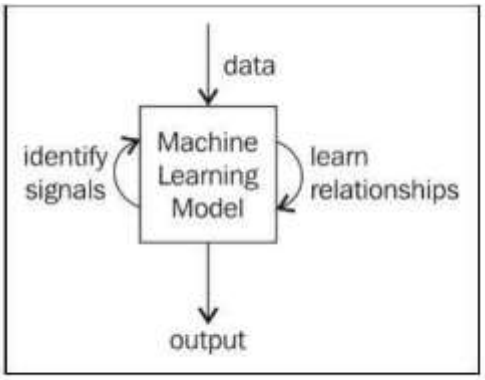


Fig 1.1.2 Traditional Programming vs Machine Learning

- Machine Learning is a combination of Algorithms, Datasets, and Programs.
- There are Many Algorithms in Machine Learning through which we will provide us the exact solution in predicting the disease of the patients.
- How Does Machine Learning Works?

- Solution to the above question is Machine learning works by taking in data, finding relationships within that data and then giving the output.



An overview of machine learning models

Fig 1.1.3 Machine Learning Model

There are various applications in which machine learning is implemented such as Web search, computing biology, finance, e-commerce, space exploration, robotics, social networks, debugging and much more.

1.1.3. EXISTING SYSTEM

In existing System, a research about a case study involving credit card fraud detection, where data normalization is applied before Naïve Bayer's and Cluster Analysis and with results obtained from the use of these methods on fraud detection has shown that by clustering attributes neuronal inputs can be minimized and promising results can be obtained by using normalized data. This research was based on unsupervised learning. Significance of this paper was to find new methods for fraud detection and to increase the accuracy of results. The data set for this paper is based on real life transactional data by a large European company and personal details in data is kept confidential. Accuracy of an algorithm is around 50%. Thus, the accuracy of the results obtained from these methods are less when compared with the proposed system.

DEMERITS OF EXISTING SYSTEM

The algorithm has few downsides such as inefficiency to handle the categorical variables which has different number of levels. Also, when there is an increase in the number of trees, the algorithm's time efficiency takes a hit.

1.1.4. PROPOSED SYSTEM

In proposed System, we are applying random forest algorithm for classification of the credit card dataset. Random Forest is an algorithm for classification and regression. Summarily, it is a collection of decision tree classifiers. Random forest has advantage over decision tree as it corrects the habit of over fitting to their training set. A subset of the training set is sampled randomly so that to train each individual tree and then a decision tree is built, each node then splits on a feature selected from a random subset of the full feature set. Even for large data sets with many features and data instances training is extremely fast in random forest and because each tree is trained independently of the others. The Random Forest algorithm has been found to provide a good estimate of the generalization error and to be resistant to over fitting.

MERITS OF PROPOSED SYSTEM

1. Random Forest output is depending on multiple decision trees which makes it un-biased, which means the end results are more reliable.

2. It is considered as robust algorithm because minute change in the dataset will not affect the end result.

3. Prevents overfitting of data.

4. Fast to train with test data.

CHAPTER-2

AIM & SCOPE

2. AIM & SCOPE

The project is mainly focused on credit card fraud detection in real world. A phenomenal growth in the number of credit card transactions, has recently led to a considerable rise in fraudulent activities. The purpose is to obtain goods without paying, or to obtain unauthorized funds from an account. Implementation of efficient fraud detection systems has become imperative for all credit card issuing banks to minimize their losses. One of the most crucial challenges in making the business is that neither the card nor the cardholder needs to be present when the purchase is being made.

2.1. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase. During system analysis the feasibility study of the proposed system is to be carried out. For feasibility analysis, some understanding of the major requirements for the system is essential.

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 system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

2.1.1. 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 equipment’s 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?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. Thus, it provides an easy access to the users. Therefore, it provides the technical guarantee of accuracy, reliability and security. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

2.1.2. Operational Feasibility:

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is their sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.1.3. Economic feasibility:

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating

the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies.

2.2. SYSTEM REQUIREMENT SPECIFICATION

A Software Requirements Specification (SRS) – a requirements specification for a software system– is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification is a structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions.

2.2.1. FUNCTIONAL REQUIREMENTS

A Functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

- Functional Requirements concerns with the specific functions delivered by the system.
- So, functional requirements are statements of the services that the system must provide.
- The functional requirements of the system should be both complete and consistent.
- Completeness means that all the services required by the user should be defined.
- Consistencymeansthatrequirementsshouldnothaveanycontradictorydefinitions.
- The requirements are usually described in a fairly abstract way. However, functional system requirements describe the system function in details, its inputs and outputs, exceptions and soon.
- Take user id and password match it with corresponding file entries. If a match is found then continue else raise an error message.

2.2.2. NON-FUNCTIONAL REQUIREMENTS

- Non-functional Requirements refer to the constraints or restrictions on the system. They may relate to emergent system properties such as reliability, response time and store occupancy or the selection of language, platform, implementation techniques and tools.
- The non-functional requirements can be built on the basis of needs of the user, budget constraints, organization policies and etc.

1. **Performance requirement:** All data entered shall be up to mark and no flaws shall be there for the performance to be 100%.
2. **Platform constraints:** The main target is to generate an intelligent system to predict the adult height.
3. **Accuracy and Precision:** Requirements are accuracy and precision of the data.
4. **Modifiability:** Requirements about the effort required to make changes in the software. Often, the measurement is personnel effort (person-months).
5. **Portability:** Since mobile phone is handy so it is portable and can be carried and used whenever required.
6. **Reliability:** Requirements about how often the software fails. The definition of a failure must be clear. Also, don't confuse reliability with availability which is quite a different kind of requirement. Be sure to specify the consequences of software failure, how to protect from failure, a strategy for error Prediction, and a strategy for correction.

7. **Security:** One or more requirements about protection of your system and its data.
8. **Usability:** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.

ACCESSIBILITY:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project people who have registered with the registration page can access their data with the help of login. User interface is simple and efficient and easy to use.

MAINTAINABILITY:

In software engineering, maintainability is the ease with which a software product can be modified in order to include new functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing project using .net and programming languages. Since the programming is very simple, it is easier to find and correct the defects and to make the changes in the project.

SCALABILITY:

System is capable of handling increase total throughput under an increased load when resources (typically hardware) are added. System can work normally under situations such as low bandwidth and large number of users.

PORTABILITY:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependent assemblies would have to be configured in such case.

VALIDATION:

It is the process of checking that a software system meets specifications and that it fulfils its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Software validation checks that the software product satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements, not as specification artifacts or as needs of those who will operate the software only; but, as the needs of all the stakeholders.

2.2.3. HARDWARE REQUIREMENTS

System	: Pentium4, IntelCorei3, i5, i7and2GHzMinimum
RAM	: Minimum 8GB
Hard Disk	: 100 GB or above
Input Device	: Keyboard and Mouse
Output Device	: Monitor or PC

2.2.4. SOFTWARE REQUIREMENTS

Operating System	: Windows 7, 10 or Higher Versions
Platform	: Anaconda Prompt (anaconda3)
Programming Language	: Python

CHAPTER - 3

CONCEPTS & METHODS

3. CONCEPTS & METHODS

3.1. PROBLEM DEFINITION

The project is mainly focused on credit card fraud detection in real world. A phenomenal growth in the number of credit card transactions, has recently led to a considerable rise in fraudulent activities. The purpose is to obtain goods without paying, or to obtain unauthorized funds from an account. Implementation of efficient fraud detection systems has become imperative for all credit card issuing banks to minimize their losses. One of the most crucial challenges in making the business is that neither the card nor the cardholder needs to be present when the purchase is being made. This makes it impossible for the merchant to verify whether the customer making a purchase is the authentic cardholder or not. With the proposed scheme, using random forest algorithm the accuracy of detecting the fraud can be improved can be improved. Classification process of random forest algorithm to analyze data set and user current dataset. Finally optimize the accuracy of the result data. The performance of the techniques is evaluated based on accuracy. Then processing of some of the attributes provided identifies the fraud detection and provides the graphical model visualization.

3.2. PROPOSED DESCRIPTION

Feature selection is an important part in machine learning to reduce data dimensionality and extensive research carried out for a reliable feature selection method. For feature selection filter method and wrapper method have been used. In filter method, features are selected on the basis of their scores in various statistical tests that measure the relevance of features by their correlation with dependent variable or outcome variable. Wrapper method finds a subset of features by measuring the usefulness of a subset of feature with the dependent variable. The classifier consider the subset of feature with which the classification algorithm performs the best. To find the subset, the evaluator uses different search techniques like depth first search, random search, breadth first search or hybrid search. The filter method uses an attribute evaluator along with a ranker to rank all the features in the dataset. Here one feature is omitted at a time that has lower ranks and then sees the predictive accuracy of the classification algorithm. Weights or rank put by the ranker algorithms are different than those by the classification algorithm. Wrapper method is useful for machine learning test whereas filter method is suitable for data mining test because data mining has thousands of millions of features.

3.2.1. ALGORITHM PROPOSED

Random Forest:

Random forest is a **Supervised Machine Learning Algorithm** that is **used widely in Classification and Regression problems.** It builds decision trees on different samples and takes their majority vote for classification.

Steps involved in Random Forest Algorithm:

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on **Majority Voting** or Averaging for Classification and regression respectively.

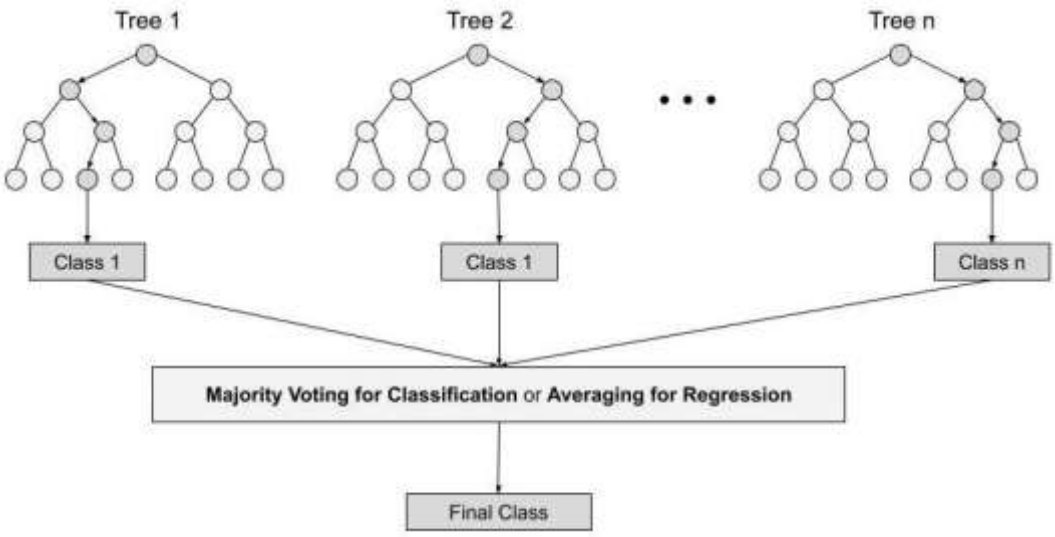


Fig 3.2.1 RANDOM FOREST DIAGRAM

3.2.2. MODULES

1. Upload credit Card Data

2. Generate Train and Test Model

3. Run Random Forest Algorithm

4. Detect fraud from the data

5. Visualization graph

First of all, we need to collect data of previous credit card transactions. As the credit card data set is not available because of privacy issues we can use the dataset with time period of two months from Kaggle website. After downloading the dataset in CSV file, Upload credit card dataset.

After Uploading dataset generate Train and Test model for Random Forest Classifier.

After generating model, we can see total records available in dataset and then application using how many records for training and how many for testing. Run Random Forest Algorithm for accuracy.

Now upload a test data contains normal and fraud transactions for prediction. It will detect fraud from the data.

Finally, a graph will appear with the transactions.

3.3. SYSTEM ANALYSIS METHODS

3.3.1. USE CASE DAIGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

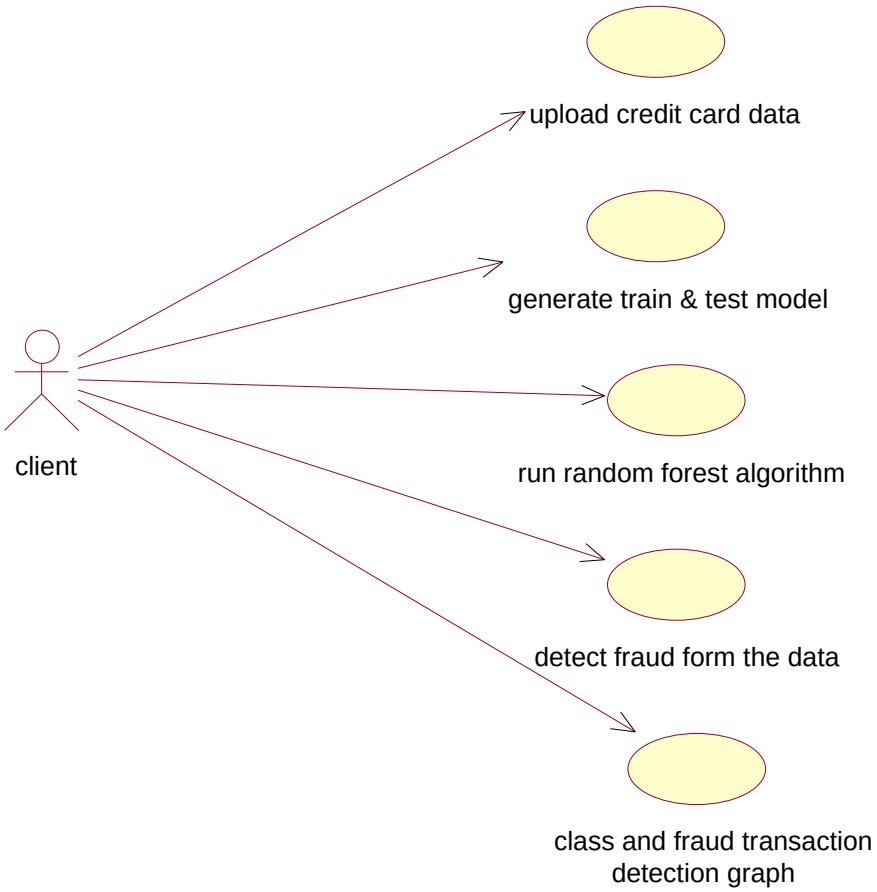


Fig: 3.3.1. Use Case Diagram

3.3.2. CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

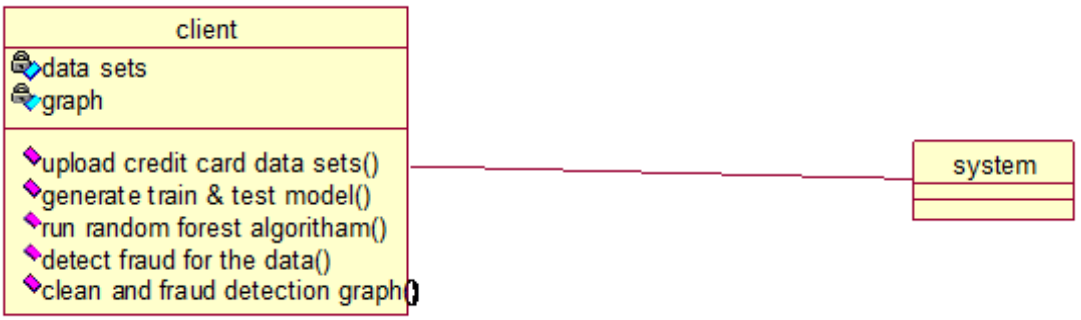


Fig 3.4.2. Class Diagram

3.3.3. SEQUENCE DIAGRAM

The Sequence diagram of the project Credit Card Fraud Detection using machine learning consist of all the various aspects a normal sequence diagram requires. This sequence diagram shows how from starting the model flows from one step to another, like he enters into the system then enters all the information's and all other general information along with the symptoms that goes into the system, compares with the prediction model and if true is predicts the appropriate results otherwise it shows the details where the user if gone wrong while entering the information's and it also shows the appropriate precautionary measure for the user to follow. Here the sequence of all the entities is linked to each other where the user gets started with the system.

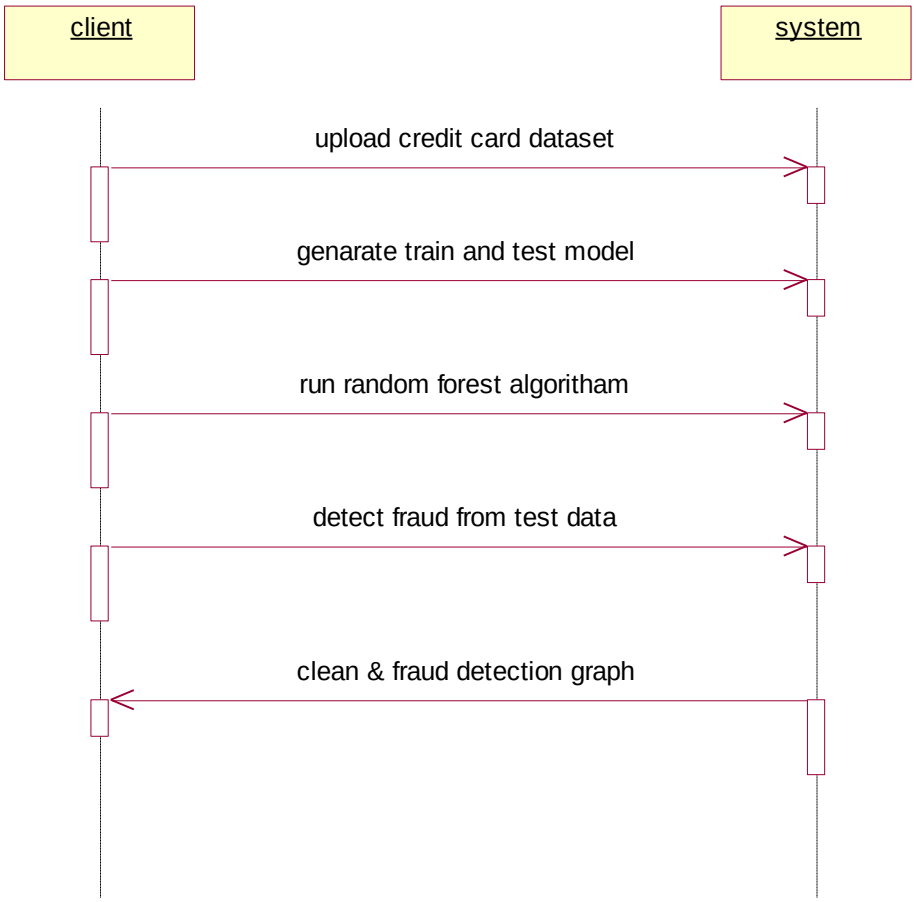


Fig 3.4.3. Sequence Diagram

CHAPTER – 4

IMPLEMENTATION

4. IMPLEMENTATION

4.1. TOOLS USED

4.1.1. ANACONDA

Anaconda Individual Edition contains conda and Anaconda Navigator, as well as Python and hundreds of scientific packages. When you installed Anaconda, you installed all these too.

Conda works on your command line interface such as Anaconda Prompt on Windows and terminal on macOS and Linux.

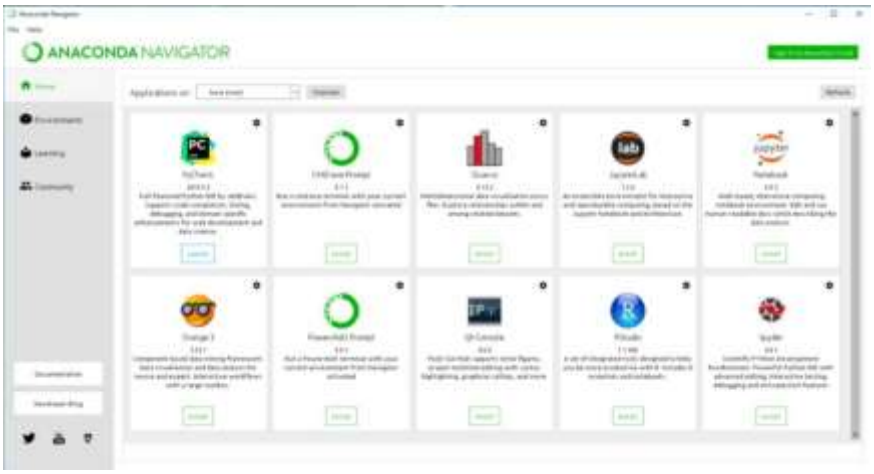
Navigator is a desktop graphical user interface that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands.

You can try both conda and Navigator to see which is right for you to manage your packages and environments. You can even switch between them, and the work you do with one can be viewed in the other.

ANACONDA NAVIGATOR 

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

To get Navigator, get the Navigator Cheat Sheet and install Anaconda. The Getting started with Navigator section shows how to start Navigator from the shortcuts or from a terminal window.



Use of Anaconda Navigator:

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

Applications of Anaconda Navigator:

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- Spyder
- PyCharm
- VSCode
- Glueviz

- Orange 3 App
- RStudio
- Anaconda Prompt (Windows only)
- Anaconda PowerShell (Windows only)

Advanced conda users can also build their own Navigator applications.

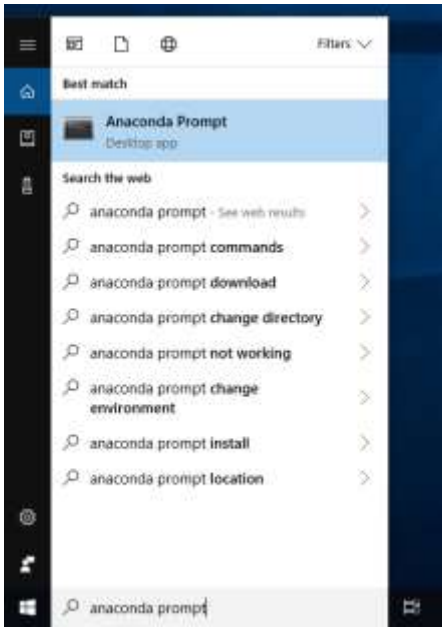
ANACONDA PROMPT

Anaconda Prompt is a command line shell (a program where you type in commands instead of using a mouse). The black screen and text that makes up the **Anaconda Prompt** doesn't look like much, but it is really helpful for problem solvers using Python.

If you prefer using a command line interface (CLI), you can use conda to verify the installation using Anaconda Prompt on Windows or terminal on Linux and macOS.

To open Anaconda Prompt:

- Windows: Click Start, search, or select Anaconda Prompt from the menu.



4.1.2. SUBLIME TEXT3

Sublime Text is a shareware cross-platform source code editor with a Python application programming interface (API).

It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

FEATURES OF SUBLIME TEXT

The following is a list of features of Sublime Text:

- "Go to Anything," quick navigation to files, symbols, or lines
- "Command palette" uses adaptive matching for quick keyboard invocation of arbitrary commands.
- Simultaneous editing: simultaneously make the same interactive changes to multiple selected areas.
- Python-based plugin API.
- Project-specific preferences.
- Extensive customizability via JSON settings files, including project-specific and platform-specific settings.
- Cross-platform (Windows, macOS, and Linux) and Supportive Plugins for cross-platform.
- Compatible with many language grammars from TextMate.

4.1.3. HEROKU CLOUD SERVICE

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js , Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages.

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku works with a wide variety of customers and partners. Learn more about how we support digital and software development agencies, partners, and enterprise companies.

4.2. PSEUDO CODE

CODE FOR HEALTH PREDICTION:

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import matplotlib.pyplot as plt
import numpy as np
from tkinter.filedialog import askopenfilename
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
#from sklearn.tree import export_graphviz
#from IPython import display


main = tkinter.Tk()

main.title("Credit Card Fraud Detection") #designing main screen

main.geometry("1300x1200")
```

```
global filename
global cls
global X, Y, X_train, X_test, y_train, y_test
global random_acc # all global variables names define in above lines
global clean
global attack
global total
def traintest(train):     #method to generate test and train data from dataset
    X = train.values[:, 0:29]
    Y = train.values[:, 30]
    print(X)
    print(Y)
    X_train, X_test, y_train, y_test = train_test_split(
    X, Y, test_size = 0.3, random_state = 0)
    return X, Y, X_train, X_test, y_train, y_test


def generateModel():              #method to read dataset values which contains all five features data
    global X, Y, X_train, X_test, y_train, y_test
    train = pd.read_csv(filename)
    X, Y, X_train, X_test, y_train, y_test = traintest(train)
    text.insert(END,"Train & Test Model Generated\n\n")
    text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")
    text.insert(END,"Split Training Size : "+str(len(X_train))+"\n")
    text.insert(END,"Split Test Size : "+str(len(X_test))+"\n")


def upload(): #function to upload tweeter profile
    global filename
    filename = filedialog.askopenfilename(initialdir="dataset")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");

def prediction(X_test, cls):  #prediction done here
    y_pred = cls.predict(X_test)
    for i in range(50):
      print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
    return y_pred
```

```
# Function to calculate accuracy

        def cal_accuracy(y_test, y_pred, details):
            accuracy = accuracy_score(y_test,y_pred)*100
            text.insert(END,details+"\n\n")
            text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
            return accuracy

        def runRandomForest():
            headers =
        ["Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V1
        5","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amo
        unt","Class"]
            global random_acc
            global cls
            global X, Y, X_train, X_test, y_train, y_test
            cls =
        RandomForestClassifier(n_estimators=50,max_depth=2,random_state=0,class_weight='balanced')
            cls.fit(X_train, y_train)
            text.insert(END,"+-+*****+Prediction Results\n\n")
            prediction_data = prediction(X_test, cls)
            random_acc = cal_accuracy(y_test, prediction_data,'Random Forest Accuracy')

#str_tree = export_graphviz(cls, out_file=None, feature_names=headers,filled=True,
special_characters=True, rotate=True, precision=0.6)
#display.display(str_tree)

        def predicts():
            global clean
            global attack
            global total
            clean = 0;
            attack = 0;
            text.delete('1.0', END)
                    filename = filedialog.askopenfilename(initialdir="dataset")
            test = pd.read_csv(filename)
            test = test.values[:, 0:29]
            total = len(test)
            text.insert(END,filename+" test file loaded\n");
```

```
    y_pred = cls.predict(test)
    for i in range(len(test)):
        if str(y_pred[i]) == '1.0':
            attack = attack + 1
            text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Contains Fraud Transaction
            Signature')+"\n\n")
        else:
            clean = clean + 1
            text.insert(END,"X=%s, Predicted = %s" % (test[i], 'Transaction Contains Cleaned
            Signatures')+"\n\n")

def graph():
    height = [total,clean,attack]
    bars = ('Total Transactions','Normal Transaction','Fraud Transaction')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, height)
    plt.xticks(y_pos, bars)
    plt.show()

font = ('times', 16, 'bold')
title = Label(main, text='Credit Card Fraud Detection Using Random Forest Tree Based Classifier')
title.config(bg='greenyellow', fg='dodger blue')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)

font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
```

```
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)

font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Credit Card Dataset", command=upload)
uploadButton.place(x=50,y=550)
uploadButton.config(font=font1)

modelButton = Button(main, text="Generate Train & Test Model", command=generateModel)
modelButton.place(x=350,y=550)
modelButton.config(font=font1)

runrandomButton = Button(main, text="Run Random Forest Algorithm",
command=runRandomForest)
runrandomButton.place(x=650,y=550)
runrandomButton.config(font=font1)

predictButton = Button(main, text="Detect Fraud From Test Data", command=predicts)
predictButton.place(x=50,y=600)
predictButton.config(font=font1)

graphButton = Button(main, text="Clean & Fraud Transaction Detection Graph", command=graph)
graphButton.place(x=350,y=600)
graphButton.config(font=font1)

exitButton = Button(main, text="Exit", command=exit)
exitButton.place(x=770,y=600)
exitButton.config(font=font1)

main.config(bg='LightSkyBlue')
main.mainloop()
```

CHAPTER -5

SCREEN SHOTS

5. SCREEN SHOTS

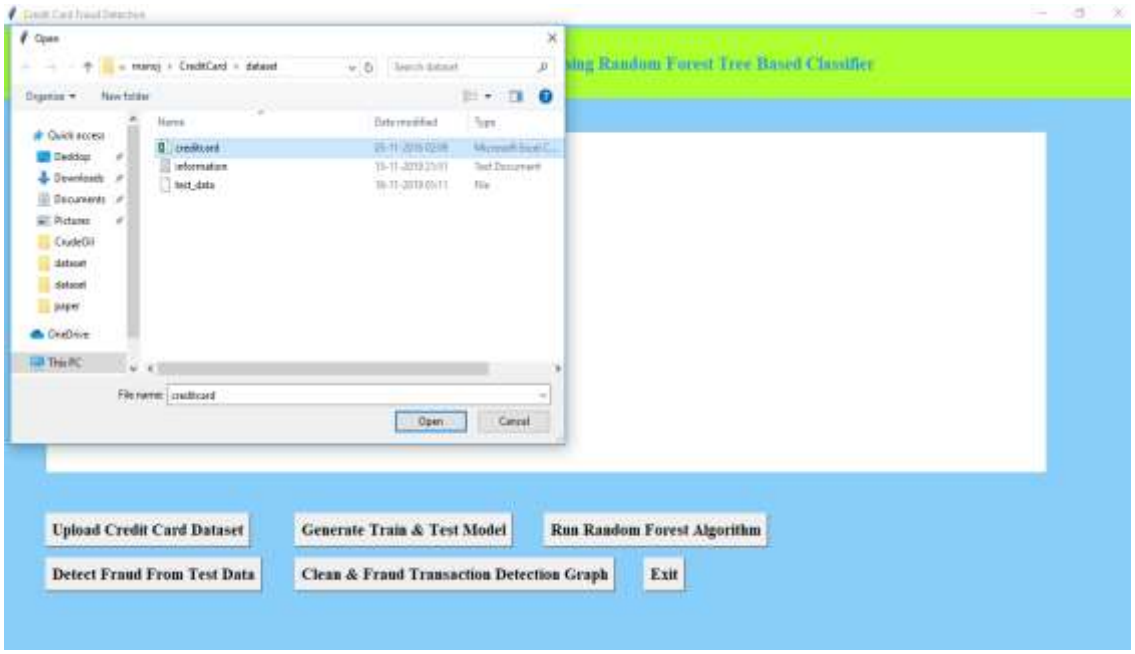


Fig: 5.1. Screenshot of Uploading credit Card Data

Fig: 5.2 Screenshot of Generating Train and Test Model

Now click on 'Generate Train & Test Model' to generate training model for Random Forest Classifier.

Fig: 5.3. Screenshot of Running Random Forest Algorithm

In above screen after generating model, we can see total records available in dataset and then application using how many records for training and how many for testing. Now click on "Run Random Forest Algorithm' button to generate Random Forest model on train and test data.

Fig: 5.4. Screenshot of obtained Accuracy

In above screen we can see Random Forest generate 99.78% percent accuracy while building model on train and test data. Now click on 'Detect Fraud from Test Data' button to upload test data and to predict whether test data contains normal or fraud transaction.

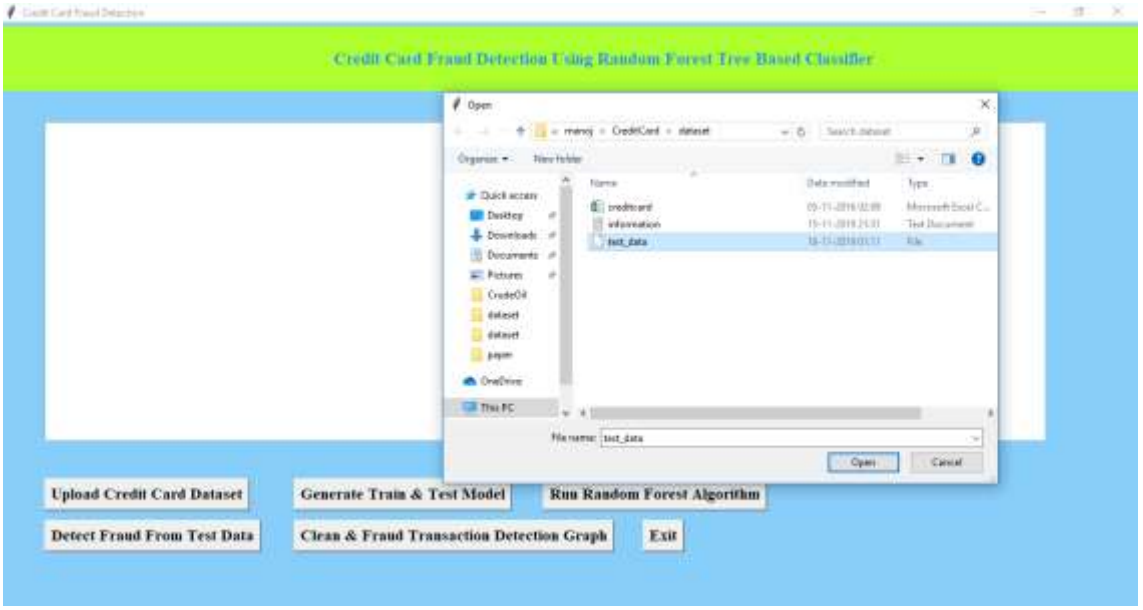


Fig: 5.5. Screenshot of Uploading Test Data

In above screen I am uploading test dataset and after uploading test data will get below prediction details.

Fig: 5.6. Screenshot of Detecting Fraud from the Data

In above screen beside each test data application will display output as whether transaction contains cleaned or fraud signatures. Now click on 'Clean & Fraud Transaction Detection Graph' button to see total test transaction with clean and fraud signature in graphical format. See the below screen,

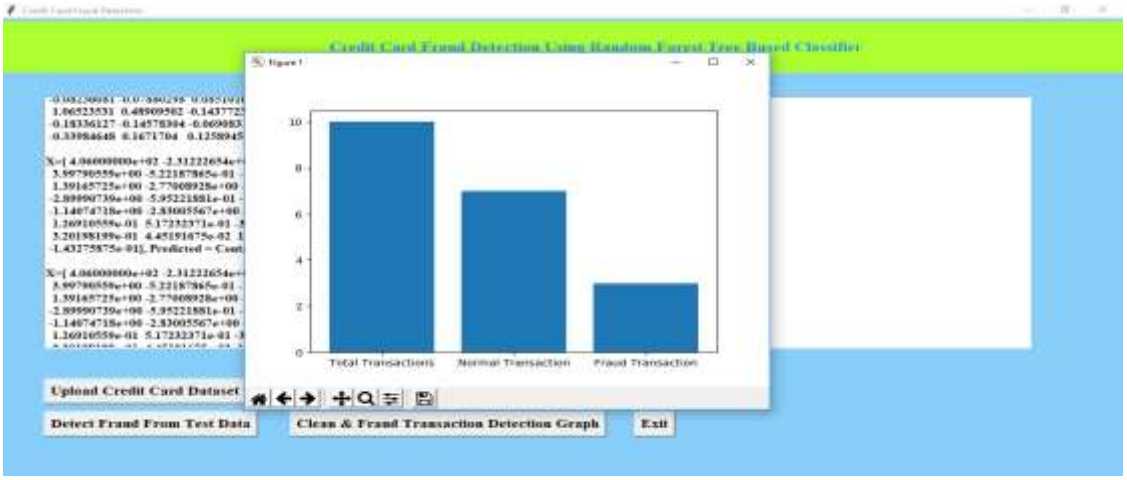


Fig: 5.7. Screenshot of Visualization Graph

In above graph we can see total test data and number of normal and fraud transaction detected. In above graph x-axis represents type and y-axis represents count of clean and fraud transaction.

CHAPTER - 6

TESTING

6. TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system.

TYPES OF TESTING

6.1. UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration.

6.2. INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.3. VALIDATION TESTING

An engineering validation test (EVT) is performed on first engineering prototypes, to ensure that the basic unit performs to design goals and specifications. It is important in identifying design problems, and solving them as early in the design cycle as possible, is the key to keeping projects on time and within budget .

Too often, product design and performance problems are not detected until late in the product development cycle — when the product is ready to be shipped. The old adage holds true: It costs a penny to make a change in engineering, amide in production and a dollar after a product is in the field.

Verification is a Quality control process that issued to evaluate whether or not a product , service, or system complies with regulations, specifications, or conditions imposed at the start of a development phase. Verification can be in development, scale-up, or production. This is often an internal process.

Validation is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

The testing process overview is as follows:

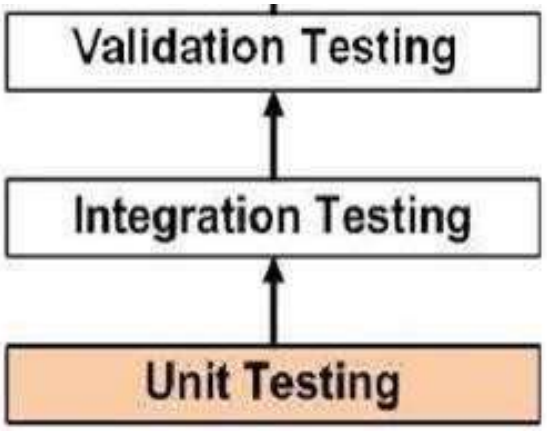


Fig 6.1 Testing Process

6.4. SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. System testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) or System Requirement Specification (SRS).

6.5. TESTING OF INITIALIZATION AND UICOMPONENTS

Serial Number of Test Case	TC 01
Module Under Test	User Registration
Description	A user enters their details for registering themselves to the System
Input	Details of Users such as username, email, phone, age.
Output	If the user's details are correct, user is registered. If the user's details are incorrect, Displays error message. If the user is already registered, Displays error message.
Remarks	Test Successful.

Table 6.5.1 Test Case for User Registration

Serial Number of Test Case	TC 02
Module Under Test	User Login
Description	When the user tries to log in, details of user are verified in the system.
Input	Username and Password.
Output	If the login details are correct, the user is logged in and user page is displayed. If the login details are incorrect, Displays error message.
Remarks	Test Successful.
Error	Check Credentials

Table 6.5.2 Test Case for User Login

Serial Number of Test Case	TC 03
Module Under Test	Prediction Result
Description	User needs to select the symptoms to get the prediction result.
Input	Name and Symptoms
Output	If user selects all correct symptoms based on their health, then the accuracy will be correct.
Remarks	Test Successful.

Table 6.5.3 Test Case for Prediction Result

CHAPTER -7

SUMMARY & CONCLUSION

7. SUMMARY & CONCLUSION

In earlier stage fraudulent cases has very less because everyone has direct cash withdrawal from bank or if any urgency they go directly. But today fraudulent cases reported day by day. Although random forest obtains good results on small set data, there are still some problems such as imbalanced data. But data should be balanced by using smote technique. The accuracy level compared to other algorithm it gives more and also choose best feature extraction and pre-processing technique to enhance the algorithm performance. Random forest algorithm performs well in handling huge amount of highly imbalanced datasets in minimum amount of time. With the accuracy of 99% in the end results it shows significant growth in detecting credit card fraud transactions when compared to the algorithms like decision tree, support vector machines and logistic regression …, because they are not performed well in handling imbalanced data sets. The Random Forest algorithm will perform better with a larger number of training data. Application of more pre-processing techniques would also help. Random forest solves the overfitting issue by using bunch of decision trees. Random forest considered as the solution for Imbalanced classification.

CHAPTER -8

FUTURE ENHANCEMENT

8. FUTURE ENHANCEMENT

The Random backwoods calculation will perform better with a bigger number of preparing information, yet speed during testing and application will endure. Utilization of more pre-preparing procedures would likewise help. The SVM calculation actually experiences the imbalanced dataset issue and requires more pre-processing to give better outcomes at the outcomes appeared by SVM is incredible however it might have been exceptional if more pre-processing have been done on the information.

CHAPTER-9

BIBLIOGRAPHY

9. BIBLIOGRAPHY

► [1] “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy”,Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, Gianluca Bontempi, IEEE on Neural Networks and Learning Systems,2018.

► [2] “A new Credit card fraud detecting method based on behavior certificate”, Lutao Zheng, Guanjnu Liu, Wenjing Luan, Zhengchuan Li, Yuwei Zhang, Chungang Yan, Changjun Jiang, 2018 IEEE 15th Internatinal Conference on Networking,Sensing and Control(ICNSC).

► [3] ”Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study”, Sahil Dhankhad ,Emad Mohammed , Behrouz Far, 2018 IEEE International Conference on Information Reuse and Integration(IRI).

► [4] "Credit Card Fraud Detection using Machine Learning Models and Collating Machine Learning models", NavanshuKhare and Saad YunusSait, International Journal of Pure and Applied Mathematics, Volume 118 No. 20 2018, 825- 838,2018.

► [5] ”Credit Card Fraud Detection using learning to Rank Approach”, N.Kalaiselvi, S.Rajalakshmi, J.Padmavathi, Joyce B.Karthiga, 2018 International Conference on Computation of Power, Energy, Information and Communication(ICCPEIC).

► [6]"Credit Card Fraud Detection using Machine Learning Techniques" John O. Awoyemi , Adebayo O. Adetunmbi, Samuel A. Oluwadare, International conference on Computing networks and informatics(ICCNI),2017.