**Credit Card Fraud Detection Using Random Forest & Cart Algorithm**

Internship Report submitted to Osmania University, Hyderabad in partial fulfilment of the requirement for the award of the degree of

# Bachelor of Engineering

in

# Computer Science and Engineering (Data Science)

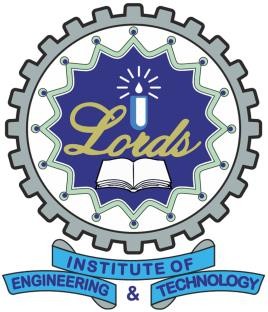
By

# Syed Saaduddin Azhaan(160922750158)

Under the Supervision of

# Mr Asif Ahmad

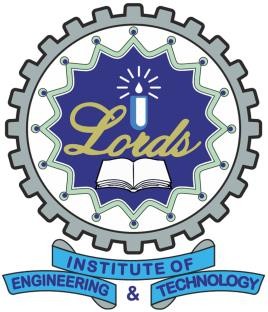
Assistant Professor



# Department of Computer Science and Engineering (Data Science) LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

**Himayathsagar, Hyderabad – 500091.**



# Department of Computer Science and Engineering (Data Science)

# LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

**(Autonomous)**

(Approved by AICTE, recognized by the Govt. of TS, and affiliated to OU) Himayathsagar, Hyderabad – 5000091.

# CERTIFICATE

This is to certify that the Internship report entitled “Credit Card Fraud Detection Using Random Forest & Cart Algorithm**”** being submitted by **Syed Saaduddin Azhaan ,** bearing **H.T.No: 160922750158** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Engineering in Computer Science and Engineering (Data Science)** is a record of bonafide work carried out by him.

**Faculty Coordinator Head of the Department Mr Asif Ahmad Mr Syed Mushtaq Ali**

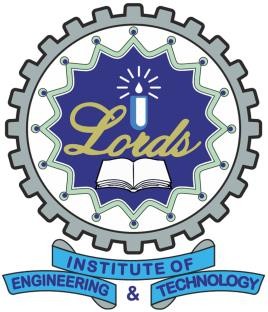
**Assistant Professor Assistance Professor & HOD,CSD**

Attach certificate soft copy

# LORDS INSTITUTE OF ENGINEERING & TECHNOLOGY

**(Autonomous) Himayathsagar, Hyderabad – 500091.**

# Department of Computer Science and Engineering(Data Science)



**DECLARATION BY THE CANDIDATE**

I**, Syed Saaduddin Azhaan,** bearing Hall Ticket No.**160922750158,** hereby declare that the Internship entitled **“**Credit Card Fraud Detection Using Random Forest & Cart Algorithm” under the guidance of **Mr Asif Ahmad (Assistant Professor)** Department of Computer Science Engineering (Data Science), **Lords Institute of Engineering & Technology, Hyderabad** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Engineering** in Computer Science and Engineering (Data Science).

This is a record of bonafide work carried out by me and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this Internship report have not been submitted to any other university or institute for the award of any other degree or diploma.

**Syed Saaduddin Azhaaan H.T.No:160922750158**

**ACKNOWLEDGMENT**

I am very pleased to present this report of my internship work. This period of my student life has been truly rewarding a number of people were of immense help to me during the course of my research and the preparation of my thesis.

First, I wish to thank **GOD Almighty** who created heavens and earth, who helped me in completing this project and I also thank my parents who encouraged me in this period of research.

I would like to thank MANAC company, for guiding and providing the opportunity to work along with them in real time.

I would like to thank **Mr Asif Ahmad Assistant Professor, Dept. of Computer Science and Engineering (Data Science) at Lords Institute of Engineering & Technology, Hyderabad**, , for his/her support and mentorship during this internship.

I would like to express my deep sense of gratitude to **Mr Syed Mushtaq Ali, Assistant Professor & HOD,** Computer Science and Engineering (Data Science), **Lords Institute of Engineering & Technology, Hyderabad,** for his encouragement and cooperation throughout the project.

I would also like to thank **Dr. R.K Singh**, **Principal** of our college, for extending his help.

**Name:Syed Saaduddin Azhaan**

**H.T.No:160922750158**

# ABSTRACT

**The primary focus of this project is on developing an effective credit card fraud detection system to address the rising incidence of fraudulent activities in the real world. With the phenomenal growth in the number of credit card transactions in recent years, there has been a significant increase in fraudulent practices, which pose serious financial risks to individuals and institutions. Credit card fraud typically involves obtaining goods or services without proper payment or unauthorized withdrawal of funds from a user’s account. Therefore, the implementation of efficient and accurate fraud detection systems has become a critical necessity for all credit card issuing banks and financial institutions in order to minimize their losses and enhance customer trust.**

**One of the most pressing challenges in combating credit card fraud is the nature of online and card-not-present (CNP) transactions. In such transactions, neither the physical card nor the cardholder is required to be present, making it impossible for the merchant to manually verify whether the customer conducting the purchase is the legitimate cardholder. This gap creates a significant vulnerability that fraudsters exploit, resulting in unauthorized transactions. To address this issue, this project proposes the development of a fraud detection system using the Random Forest algorithm.**

**The Random Forest algorithm, a widely-used machine learning technique, leverages its ensemble classification methodology to improve the accuracy and reliability of fraud detection. By applying this algorithm to a dataset containing both legitimate and fraudulent transactions, the classification process can analyze patterns, detect anomalies, and identify potentially fraudulent activities. Additionally, the system incorporates real-time processing of the current user dataset to enhance detection capabilities. The primary goal is to optimize the accuracy of the results and ensure the system is capable of distinguishing between genuine and fraudulent transactions effectively.**

**The performance of the proposed fraud detection system is evaluated using various metrics, including accuracy, sensitivity, specificity, and precision. These metrics provide a comprehensive assessment of the model's ability to identify fraudulent transactions while minimizing false positives and false negatives. Furthermore, by processing certain key attributes in the dataset, the system is capable of generating visualizations, such as graphical models, to provide insights into the underlying data and aid in understanding the detection process.**

**In conclusion, this project aims to deliver a robust and efficient credit card fraud detection system utilizing advanced machine learning techniques. By improving the accuracy of fraud detection and offering insightful visualizations, the proposed solution can contribute significantly to mitigating financial losses for credit card issuers and enhancing the security of credit card transactions.**

# LIST OF UML DIAGRAMS

|  |  |  |
| --- | --- | --- |
| SNO: | NAME OF THE DIAGRAM | PAGE NO |
| 1. | **Class diagram** | 21 |
| 2. | **Use case Diagram** | 21 |
| 3. | **Sequence Diagram** | 22 |
| 4. | **Collaboration Diagram** | 22 |

**INDEX**

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
| Department Certificate | ii |
| Company Certificate | iii |
| Declaration | iv |
| Acknowledgement | v |
| Abstract | vi |
| List of Diagrams | vii |

|  |  |
| --- | --- |
| * CHAPTER-1: INTRODUCTION | 9 |
| * CHAPTER-2: LITERATURE SURVEY | 11 |
| * CHAPTER-3: INPUT AND OUTPUT DESIGN | 13 |
| * CHAPTER-4: SYSTEM ANALYSIS | 15 |
| * CHAPTER-5: REQUIREMENT SPECIFICATIONS | 17 |
| * CHAPTER-6: SYSTEM STUDY | 18 |
| * CHAPTER-7: SYSTEM ARCHITECTURE | 20 |
| * CHAPTER-8:SYSTEM MODULES | 21 |
| * CHAPTER-9: UML DIAGRAMS | 24 |
| * CHAPTER-10:PROJECT EXECUTION AND TESTING | 26 |
| * CHAPTER-11: CONCLUSION | 36 |
| * CHAPTER-12:BIBLOGRAPHY | 37 |

# CHAPTER 1

# INTRODUCTION

Various techniques for detecting fraudulent activities in credit card transactions have been proposed and implemented, with researchers exploring methods based on artificial intelligence, data mining, fuzzy logic, and machine learning. Credit card fraud detection is not only a challenging problem but also a highly popular and critical one to solve in the current era of digital finance. With our proposed system, we aim to tackle this issue by building an advanced credit card fraud detection model using machine learning.

Machine learning has emerged as a highly successful approach for fraud detection due to its ability to handle vast amounts of transactional data and detect patterns that are often invisible to human analysts. Online credit card transactions generate a significant volume of data, leading to binary outcomes of either genuine or fraudulent transactions. Within these datasets, features are constructed to provide insights into the behavior of transactions. These features include data points such as the age of the customer account, the value of transactions, the origin of the credit card, and many more. In fact, datasets can consist of hundreds of features, each contributing to the overall fraud probability.

The contribution of each feature to the fraud probability is determined by the machine learning model itself, driven by the training dataset. Unlike traditional methods where fraud analysts manually assign weights to features, machine learning models self-learn to make these determinations without explicit programming. This adaptability ensures that as patterns in fraud evolve, the model can adjust accordingly. For instance, if fraudulent activity involving a specific credit card type increases, the model will assign a higher fraud weight to transactions using that card type. Conversely, if the activity decreases, the contribution level will adjust in parallel, showcasing the model’s dynamic nature.

Our system deploys machine learning models that utilize both classification and regression algorithms to detect fraudulent activities. Specifically, we focus on supervised learning techniques, with the Random Forest algorithm being a key component of our approach. Random Forest is an advanced and robust version of the Decision Tree algorithm, known for its superior accuracy and efficiency compared to other machine learning methods.

The Random Forest algorithm addresses challenges such as correlation issues by selecting only a random subset of features at each node split, thereby decorrelating individual trees. Additionally, the algorithm employs a pruning mechanism by setting a stopping criterion for node splits, which ensures the trees remain efficient and do not overfit the data. By leveraging these techniques, Random Forest builds an ensemble of decision trees, which collectively produce highly accurate predictions. This makes it a particularly effective tool for classifying credit card transactions as either fraudulent or legitimate in both online and offline scenarios.

In conclusion, the integration of machine learning, and specifically the Random Forest algorithm, into credit card fraud detection offers a powerful and scalable solution. By continuously learning from new data and adapting to evolving patterns, our system can significantly enhance the detection of fraudulent activities. This not only helps minimize financial losses for financial institutions but also builds trust and security among credit card users in an increasingly digital world.

# CHAPTER-2

# LITERATURE SURVEY

### The Use of Predictive Analytics Technology to Detect Credit Card Fraud in Canada. “Kosemani Temitayo Hafiz, Dr. Shaun Aghili, Dr. Pavol Zavarsky.”

This research paper focuses on the creation of a scorecard from relevant evaluation criteria, features, and capabilities of predictive analytics vendor solutions currently being used to detect credit card fraud. The scorecard provides a side-by- side comparison of five credit card predictive analytics vendor solutions adopted in Canada. From the ensuing research findings, a list of credit card fraud PAT vendor solution challenges, risks, and limitations was outlined.

### BLAST-SSAHA Hybridization for Credit Card Fraud Detection. “Amlan Kundu, Suvasini Panigrahi, Shamik Sural, Senior Member, IEEE, and Arun K. Majumdar”

This paper propose to use two-stage sequence alignment in which a profile Analyser (PA) first determines the similarity of an incoming sequence of transactions on a given credit card with the genuine cardholder’s past spending sequences. The unusual transactions traced by the profile analyser are next passed on to a deviation analyser (DA) for possible alignment with past fraudulent behaviour. The final decision about the nature of a transaction is taken on the basis of the observations by these two analysers. In order to achieve online response time for both PA and DA, we suggest a new approach for combining two sequence alignment algorithms BLAST and SSAHA.

### Research on Credit Card Fraud Detection Model Based on Distance Sum.

**“Wen-Fang YU, Na Wang”.**

Along with increasing credit cards and growing trade volume in China, credit card fraud rises sharply. How to enhance the detection and prevention of credit card fraud becomes the focus of risk control of banks. It proposes a credit card fraud detection model using outlier detection based on distance sum according to the infrequency and unconventionality of fraud in credit card transaction data, applying outlier mining into credit card fraud detection. Experiments show that this model is feasible and accurate in detecting credit card fraud.

### Fraudulent Detection in Credit Card System Using SVM & Decision Tree. “Vijayshree B. Nipane, Poonam S. Kalinge, Dipali Vidhate, Kunal War, Bhagyashree P. Deshpande”.

With growing advancement in 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.

### 5] Supervised Machine (SVM) Learning for Credit Card Fraud Detection. “Sitaram patel, Sunita Gond”.

This thesis propose the SVM (Support Vector Machine) based method with multiple kernel involvement which also includes several fields of user profile instead 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.

### [6] Detecting Credit Card Fraud by Decision Trees and Support Vector Machines. “Y. Sahin and E. Duman”

In this study, classification models based on decision trees and support vector machines (SVM) are developed and applied on credit card fraud detection problem. This study is one of the firsts to compare the performance of SVM and decision tree methods in credit card fraud detection with a real data set.

**CHAPTER-3**

**INPUT AND OUTPUT DESIGN**

**Input Design**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**Objectives**

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3.When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**Output Design**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2.Select methods for presenting information.

3.Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**CHAPTER-4**

## SYSTEM ANALYSIS

**Existing System**

In existing System, a research about a case study involving credit card fraud detection, where data normalization is applied before Cluster Analysis and with results obtained from the use of Cluster Analysis and Artificial Neural Networks on fraud detection has shown that by clustering attributes neuronal inputs can be minimized. And promising results can be obtained by using normalized data and data should be MLP trained. 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%. Significance of this paper was to find an algorithm and to reduce the cost measure. The result obtained was by 23% and the algorithm they find was Bayes minimum risk

**Disadvantages**

In this paper a new collative comparison measure that reasonably represents the gains and losses due to fraud detection is proposed.

A cost sensitive method which is based on Bayes minimum risk is presented using the proposed cost measure.

**Proposed Scheme**

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.

**Advantages Of Proposed System**

Random forest ranks the importance of variables in a regression or classification problem in a natural way can be done by Random Forest.The 'amount' feature is the transaction amount. Feature 'class' is the target class for the binary classification and it takes value 1 for positive case (fraud) and 0 for negative case (not fraud).

**CHAPTER-5**

**REQUIREMENT SPECIFICATIONS**

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals.

### Hardware Requirements

* + - Processor - Intel
    - RAM - 4 Gb
    - Hard Disk - 260 GB
    - Key Board - Standard Windows Keyboard
    - Mouse - Two or Three Button Mouse

### Software Requirements

Python Anaconda

OS - Windows 7, 8 and 10 (32 and 64 bit)

**CHAPTER-6**

**SYSTEM STUDY**

**Feasibility Study**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

**Three key considerations involved in the feasibility analysis are,**

* **Economical Feasibility**
* **Technical Feasibility**
* **Social Feasibility**

**Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**CHAPTER-7**

**SYSTEM ARCHITECTURE**

First the credit card dataset is taken from the source and cleaning and validation is performed on the dataset which includes removal of redundancy, filling empty

spaces in columns, converting necessary variable into factors or classes then data is divided into 2 part, one is training dataset and another one is test data set. Now the original sample is randomly partitioned into teat and train dataset.

A diagram of a process

Description automatically generated

**CHAPTER-8**

SYSTEM MODULES

**Module Description**

**Module 1: Data Collection**

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

### Module 2: Data Pre-Processing

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be removed from the data entirely.

Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

### Module 3: Feature Extration

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

## Module 4: Evaluation Model

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. **Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

## Algorithm Utilized

**Random Forest**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

## Working Of Random Forest

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

## Advantages Of Using Random Forest

Pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.

The random forest algorithm also works well when data has missing values or it has not been scaled well

**CHAPTER-9**

**UML DIAGRAMS**

Class Diagram



Use Case Diagram



Sequence Diagram:



Collaboration Diagram



**CHAPTER-10**

**PROJECT EXECUTION AND TESTING**

In this project we are using python Random Forest inbuilt Cart algorithm to detect fraud transaction from credit card dataset, we downloaded this dataset from ‘kaggles’ web site from below URL

Dataset URL: <https://www.kaggle.com/mlg-ulb/creditcardfrau>

To provide privacy to users transaction data kaggles peoples have converted transaction data to numerical format using PCA Algorithm. Below are some example from dataset

"Time","V1","V2","V3","V4","V5","V6","V7","V8","V9","V10","V11","V12","V13","V14","V15","V16","V17","V18","V19","V20","V21","V22","V23","V24","V25","V26","V27","V28","Amount","Class"

0,-1.3598071336738,-0.0727811733098497,2.53634673796914,1.37815522427443,-0.338320769942518,0.462387777762292,0.239598554061257,0.0986979012610507,0.363786969611213,0.0907941719789316,-0.551599533260813,-0.617800855762348,-0.991389847235408,-0.311169353699879,1.46817697209427,-0.470400525259478,0.207971241929242,0.0257905801985591,0.403992960255733,0.251412098239705,-0.018306777944153,0.277837575558899,-0.110473910188767,0.0669280749146731,0.128539358273528,-0.189114843888824,0.133558376740387,-0.0210530534538215,149.62,"0"

0,1.19185711131486,0.26615071205963,0.16648011335321,0.448154078460911,0.0600176492822243,-0.0823608088155687,-0.0788029833323113,0.0851016549148104,-0.255425128109186,-0.166974414004614,1.61272666105479,1.06523531137287,0.48909501589608,-0.143772296441519,0.635558093258208,0.463917041022171,-0.114804663102346,-0.183361270123994,-0.145783041325259,-0.0690831352230203,-0.225775248033138,-0.638671952771851,0.101288021253234,-0.339846475529127,0.167170404418143,0.125894532368176,-0.00898309914322813,0.0147241691924927,2.69,"0"

406,-2.3122265423263,1.95199201064158,-1.60985073229769,3.9979055875468,-0.522187864667764,-1.42654531920595,-2.53738730624579,1.39165724829804,-2.77008927719433,-2.77227214465915,3.20203320709635,-2.89990738849473,-0.595221881324605,-4.28925378244217,0.389724120274487,-1.14074717980657,-2.83005567450437,-0.0168224681808257,0.416955705037907,0.126910559061474,0.517232370861764,-0.0350493686052974,-0.465211076182388,0.320198198514526,0.0445191674731724,0.177839798284401,0.261145002567677,-0.143275874698919,0,"1"

Above bold names are the column names of this dataset and others decimal values are the content of dataset and in above 3 rows last column contains class label where 0 means transaction values are normal and 1 means contains fraud values.

Using above ‘CreditCardFraud.csv’ file we will train Random Forest algorithm and then we will upload test data file and this test data will be applied on Random Forest train model to predict whether test data contains normal or fraud transaction signatures. When we upload test data then it will contains only transaction data no class label will be there application will predict and give the result. See below test data file

A screenshot of a computer

Description automatically generated

In above screen in test data file there are no 0 or 1 values, application will predict from this test data using random forest and give the result.

Random Forest Algorithm

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance. Python SKLEARN inbuilt contains support for CART with all decision trees and random forest classifier.

Random forests has a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

To run project double click on ‘run.bat’ file to get below screen

A screenshot of a computer

Description automatically generated

In above screen click on ‘Upload Credit Card Dataset’ button to upload dataset

A screenshot of a computer

Description automatically generated

After uploading dataset will get below screen

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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 below screen

A screenshot of a computer

Description automatically generated

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

**SYSTEM TEST**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### Types Of Tests

**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. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**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.

**Functional Test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CHAPTER-11**

**CONCLUSION**

The Random Forest algorithm demonstrates improved performance when trained on a larger dataset. The increased volume of training data allows the model to capture more intricate patterns and relationships within the data, leading to higher accuracy in detecting fraudulent transactions. However, this comes with a trade-off: as the size of the training dataset grows, the speed of testing and application during real-time transactions can decrease. Therefore, optimizing the balance between training data size and operational efficiency is crucial to ensure practical implementation. Additionally, the application of advanced pre-processing techniques can significantly enhance the model’s performance by cleaning and transforming the data to highlight relevant features and eliminate noise.

On the other hand, the Support Vector Machine (SVM) algorithm, though effective, continues to face challenges when dealing with imbalanced datasets—a common issue in credit card fraud detection where fraudulent transactions constitute a very small fraction of the total data. SVM's performance is highly sensitive to data preprocessing, which includes tasks like oversampling the minority class, undersampling the majority class, or creating synthetic samples using techniques such as SMOTE (Synthetic Minority Over-sampling Technique). While the results achieved by SVM in fraud detection are promising, there is potential for further improvement. With more robust preprocessing strategies, SVM could achieve better balance and accuracy, addressing the class imbalance problem more effectively.

In summary, while Random Forest and SVM are both powerful algorithms for fraud detection, their effectiveness depends significantly on the quality of preprocessing and the structure of the dataset. Random Forest benefits from larger datasets but may face operational challenges, while SVM requires careful handling of class imbalance to optimize its results. Future implementations should focus on refining preprocessing techniques and selecting algorithms that balance accuracy and computational efficiency to achieve optimal outcomes.

**CHAPTER-12**

**BIBLOGRAPHY**

1. Sudhamathy G: Credit Risk Analysis and Prediction Modelling of Bank Loans Using R, vol. 8, no-5, pp. 1954-1966.
2. LI Changjian, HU Peng: Credit Risk Assessment for ural Credit Cooperatives based on Improved Neural Network, International Conference on Smart Grid and Electrical Automation vol. 60, no. - 3, pp 227-230, 2017.
3. Wei Sun, Chen-Guang Yang, Jian-Xun Qi: Credit Risk Assessment in Commercial Banks Based On Support Vector Machines, vol.6, pp 2430-2433, 2006.
4. Amlan Kundu, Suvasini Panigrahi, Shamik Sural, Senior Member, IEEE, “BLAST-SSAHA Hybridization for Credit Card Fraud Detection”, vol. 6, no. 4 pp. 309-315, 2009.
5. Y. Sahin and E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines, Proceedings of International Multi Conference of Engineers and Computer Scientists, vol. I, 2011.
6. Sitaram Patel, Sunita Gond , “Supervised Machine (SVM) Learning for Credit Card Fraud Detection, International of engineering trends and technology, vol. 8, no. -3, pp. 137- 140, 2014.
7. Snehal Patil, Harshada Somavanshi, Jyoti Gaikwad, Amruta Deshmane, Rinku Badgujar," Credit Card Fraud Detection Using Decision Tree Induction Algorithm, International Journal of Computer Science and Mobile Computing, Vol.4 Issue.4, April- 2015, pg. 92-95
8. Dahee Choi and Kyungho Lee, “Machine Learning based Approach to Financial Fraud Detection Process in Mobile Payment System", vol. 5, no. - 4, December 2017, pp. 12-24.