CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING

A PROJECT REPORT

Submitted by

PANDI KAVYA (19101074)

THATHIREDDY PUSHPALATHA (19101113)

YEGIREDDY DEEKSHITHA (19101124)

In partial fulfilment for the award of the degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

VIVEKANANDHA COLLEGE OF ENGINEERING FOR WOMEN

[Autonomous]

Approved by AICTE, New Delhi and Accredited by NBA [CSE, EEE, IT&BT]

Affiliated to Anna University, Chennai-25, An ISO 9001-2015 Certified Institution

Elayampalayam, Tiruchengode, Namakkal Dt. – 637205

APRIL 2023

VIVEKANANDHA COLLEGE OF ENGENEERING FOR WOMEN

[Autonomous]

Approved by AICTE, New Delhi and Accredited by NBA [CSE, EEE, IT&BT]

Affiliated to Anna University, Chennai-25, An ISO 9001-2015 Certified Institution

Elayampalayam, Tiruchengode, Namakkal Dt. – 637205

BONAFIDE CERTIFICATE

Certified that this project report **"CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING",** is the bonafide work of **"Pandi Kavya (19101074), Thathireddy Pushpalatha (19101113), Yegireddy Deekshitha (19101124)"** who carried out the project work under my supervision.

SIGNATURE OF THE HOD

Dr. C. POONGODI, M.E.,Ph.D.,
Professor & Head,
Department of Computer Science
and Engineering
Vivekanandha College of Engineering
For Women(Autonomous)

SIGNATURE OF THE SUPERVISOR

Dr. R.NITHYA, M.E.,Ph.D.,ASP/CSE
Associate Professor,
Department of Computer Science
and Engineering
Vivekanandha College of Engineering
For Women(Autonomous)

Submitted to the Viva voce Examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We thank and praise the Lord Almighty for providing us knowledge, strength and all the necessary facilities to do this report successfully.

We are immensely grateful to our honourable **Chairman and Secretary, "Vidya Rathna" Professor Dr. M. KARUNANITHI, B.Pharm., M.S., Ph.D., D.Litt.,** Vivekanandha Educational Institutions, who is our inspiration.

We are extremely grateful to our beloved **"Executive Director" Prof. Dr. S.KUPPUSWAMI, B.E., M.Sc. (ENG)., Dr.Ing (France).,** for his motivation and guidance for our final year project.

We wish to express our profound thanks to our beloved **Principal, Dr.KCK.VIJAYAKUMAR, M.E., Ph.D., MIE,** for all the facilities and support provided during the period of our final year project.

We would like to acknowledge our **Head of the Department Dr.C.POONGODI, M.E., Ph.D.,** Department of Computer Science and Engineering, for her encouragement and support for completing the projectsuccessfully.

We wish to thank our Project Coordinators **Ms.K.SIVAPRIYA., M.E., AP/CSE.,** and **Ms.A.SARANYA, M.E., AP/CSE** for their kind support and guidance in completion of our project.

We wish to thank our project faculty guide **Dr.R.Nithya, M.E.,Ph.D., ASP/CSE** for her kind support and encouragement throughout the project.

We are thankful and fortunate enough to get constant encouragement, support and guidance for our parents and all Teaching and non-Teaching staff members of the department of Computer Science and Engineering who helped us in successfully completing our project.

ABSTRACT

Credit card fraud is a growing problem that affects both financial institutions and customers worldwide. Despite the integration of chip cards and existing protection systems, new types of fraud continue to emerge, making it difficult to track and create rules in time. To address this problem, artificial intelligence techniques such as support vector machines (SVMs) have been applied to develop credit card fraud detection systems. This paper proposes a credit card fraud detection system using SVMs and describes the various modules necessary for the system's functioning. These modules include data preprocessing, feature engineering, SVM model training, real-time fraud detection, model evaluation, and reporting. Additionally, it describes the importance of input data and feature extraction, highlighting their critical roles in the credit card fraud detection process. The proposed system's advantages include its ability to detect fraudulent transactions accurately, its efficiency in processing real-time data, and its potential to reduce the number of employees required to track and prevent fraud. Overall, this paper demonstrates the effectiveness of using SVMs to develop credit card fraud detection systems, which have significant potential in enhancing financial security for institutions and customers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYMS	ABBREVIATIONS
CD	Communal Detection
SD	Spike Detection
NB	Naïve Bayes
LR	Logistic Regression
DM	Data Mining
DOB	Data-Of-Birth
FDS	Fraud Detection System
OPV	Online Pin Verification
EMV	Europay Mastercard Visa
DBMS	Database Management System
CTPOS	Card Terminal at Point of Sales
CVM	Cardholder Verification Method
SSN	Social Security Number
PIN	Personal Identification Number
CIB	Card Issuing Bank
NDA	Non-Disclosure Agreement
POS	Point-Of-Sale
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network

CHAPTER 1

INTRODUCTION

1.1 CREDIT CARD

The use of credit cards is prevalent in modern day society. But it is obvious that the number of credit card fraud cases is constantly increasing in spite of the chip cards worldwide integration and existing protection systems. This is why the problem of fraud detection is very important now. In this work the general description of the developed fraud detection system and comparisons between models based on using of artificial intelligence are given. In the last section of this work the results of evaluative testing and corresponding conclusions are considered. The use of credit cards is prevalent in modern day society. But as in other related fields, financial fraud is also occurring in spite of the chip cards worldwide integration and existing protection systems. This is why most software developers are trying to improve existing methods of fraud detection in processing systems. The majority of such methods are rules-based models. Such models allow bank employees to create the rules describing transactions that are suspicious. But the number of transactions per day is large and new types of the fraud appear quickly.

Therefore, it is very difficult to track new types of fraud and to create corresponding rules in time. It would require a significant increase in the number of employees. Such problems can be avoided using of artificial intelligence. But this task is very special and complex models are not acceptable because of authorization time limits. The use of Bayesian Networks is suitable for this type of detection, but results from previous research showed that some input data (attributes of transaction) representation

method should be used for effective classification. For transaction monitoring by bank employees the clustering model was developed. This model allows provision of fast analysis of transactions by attributes.

1.1.1 FRAUD DETECTION SYSTEM

In this system two modules (FDS ONLINEP and FDS OFFLINEP) for fraud detection (transaction classification) are used. The FDS ONLINEP module is used for on-line fraud detection, i.e., fraud detection process during authorization of transactions in a bank processing system.

In this module different models for the fraud detection can be used. If a transaction is recognized as fraudulent, in this module, then a corresponding message will be sent to the processing system and this transaction can be declined. Classification process takes some time and the time of transaction authorization is limited. This is why some models cannot be applied in the FDS ONLINEP module because of exceeding time limits. These models are used in the FDS OFFLINE module. FDS OFFLINE module allows the system to detect fraud among transactions that have been already authorize and were classified in the FDS ONLINEP module. For the storage of incoming transactions, statistical data for corresponding models, results of classification and generic parameters a FDS Data Warehouse is used. Module FDS ALERT is used for alerting credit card holders in case of fraud recognition by the FDS ONLINEP module using SMS or email messages.

The evaluative testing of two sets of transactions have been generated. The first set of transactions was generated for the training process and the second set is for the testing process. The training data consists of 83 transactions, where 52 of them correspond to legal transactions while 31 correspond to fraudulent transactions. All transactions have been generated

for one card and these transactions correspond to real using of credit card in one of the banks during three months. This set contains 21 legal purchases that have been generated using 8 different merchants through 8 different financial organizations. From these transactions 16 have been done with the" local country" parameter using local currency and other have been done with" foreign country" parameter using foreign currency. Other part of legal transactions consists of cash withdrawals. In this part 24 transactions have been generated using 5 different cash point machines of issuer bank and 7 cash withdrawals were done through cash point machines of foreign banks. The fraudulent transactions have been generated with a specific financial institution, country code and time. These transactions were generated on the base of real fraudulent transaction. The testing data consists of 9 transactions with attributes that have been observed in the legal transactions from the training data and 2 transactions that correspond to fraud. Three different tests were conducted using these sets. The first test was intended to receive a comparison between the Naive Bayesian Classifier based on the normal distribution, the discrete distribution, the kernel density estimation and the developed input data representation method. In these pictures Plegal - the probability that transaction is legal, Pfraud - the probability that transaction is fraudulent. The results of the Naïve Bayes method classification using the normal distribution, the kernel density estimation and the discrete distribution are not acceptable for this type of fraud detection. The most legal probabilities for the Naive Bayesian Classifier using these probability estimation methods are too low and are therefore incorrect.

The reasons for this for each method are different. For the normal distribution, these probabilities are low because the real distribution of values from the training set does not correspond to the normal distribution for each attribute. The reason for this for the discrete distribution and the kernel

density estimation is that a value of probability for each attribute depends on a number of different values for this attribute which have been observed in the training set. The problem is that if a number of different values for some attribute in the training set is increased then the dispersion of the attribute values will grow upwards. Therefore, the probabilities for values that hit into a center of the real distribution for corresponding attribute will be high only. But the values that have been observed in the training data may do not hit into a center of the real distribution. Consequently, the probabilities for these values will be low which is not acceptable for this type of detection. Also, the assumption about conditional independence of different attributes has a bad effect on the final probability calculation, because of according this assumption, probabilities for different attributes are multiplied. Thus, the Naive Bayesian Classifier using the normal distribution, the discrete distribution and the kernel density estimation method is not suitable for this type of detection. But results of the classification using the Naive Bayesian Classifier based on the developed input representation method and the discrete distribution for probability estimation are acceptable. In this case only two values for input attributes are possible (0 and 1) and this is why probabilities calculated using the discrete distribution are correct for this testing. This is the main reason that legal transactions have been classified as legal and fraudulent transaction.

1.2 EMV (EUROPAY MASTERCARD VISA)

EMV (Europay Master Card Visa) is a globally accepted standard for chip card-based payment transactions, which benefits from the intrinsic security characteristics of chip cards. The EMV specification is relatively flexible and can be deployed in both online and offline card acceptance environments. In the offline environment, payment terminals and cards only

communicate with each other in order to approve/decline the payment transactions, whereas in the online environment authorization entities are also involved in the overall process. An authorization entity can either be the Card Issuing Bank (CIB) or the payment scheme operator (e.g. Visa, Master-Card). Aside from the transaction authorization, the EMV specifications define offline-PIN verification as one of the main cardholder verification methods. However, in an online authorization environment, the PIN verification process is referred to as Online-PIN Verification (OPV).

This process is the main focus of this work. We discuss the OPV process that has placed indelible trust assumptions on the intermediary entities (subcontractors) between a payment terminal and a scheme operator/CIB. When this trust (assumption) is scrutinized, there is a potential attack scenario that an adversary can use to gain access to PIN data. This information can be used by an adversary to carry out an online PIN approved transaction without the involvement of the genuine cardholder but with the correct PIN. We then propose three solutions based on the existing OPV process as potential countermeasures that are then implemented to measure any incurred performance penalties. At the Card Terminal at Point of Sales (CTPOS)3, the cardholder inserts her smart card into a card terminal, after which the card and the terminal communicate with each other and try to determine common parameters based on their individual risk assessments in order to perform the transaction. If the common parameters require cardholder authentication, then the terminal will ask the consumer to enter her PIN. The input value can be validated either by the smart card or by the authorization entity (i.e. scheme operator or CIB). If the PIN is verified successfully then the card-holder is authenticated and the transaction can proceed to the next step. Knowledge of the PIN is regarded as an authorization for the transaction from the

cardholder. It is an alternative to the cardholder's signature used to authorize a payment to the merchant.

In the OPV process, between the CTPOS and the scheme operator/card issuing bank, there might be a number of entities that handle the communication. The EVM Card Specification Book 4 recommends that this communication should be adequately protected. Understanding the OPV process deployed in the ATM transactions and elaborated in, it is clear that the payment terminal requests the PIN from the cardholder, then encrypts it and sends it to the authorization entity. that it shares with the next point of communication, which might not be the authorization entity. The next point deploys a key translation mechanism and forwards the message to the next stage in the journey to the authorization entity. Furthermore, the transaction authorization message4 generated by the card, whose purpose is to get an online transaction approval from the authorization entity, has no binding with the OPV process. This message consists of a number of EMV tags that are then encrypted using a shared key (between the card and the authorization entity). One of the tags that make up the ARQC is the Cardholder Verification Method (CVM) mentioned in EMV Book 4, which indicates what method is used to verify the cardholder [6]. The CVM is a three-byte tag, with each byte representing the CVM performed, CVM conditions and CVM results [6, see: p49]. The only information in the CVM bytes regarding the OPV is a single binary value set to 1 if CVM was performed.

In the CVM or ARQC, there is no tag that binds the OPV process and the respective ARQC. Also, during the OPV, only the payment terminal handles the PIN and the card is not informed of the PIN value entered on the terminal. Therefore, if an adversary compromise one of the intermediate entities that perform the key translation between the payment terminal and the

authorization entity, he can obtain the PIN number along with other transaction details. To perform this attack, the adversary will observe all the OPV messages that include the PINs and associated Primary Account Numbers (PAN). This will enable the adversary to perform the OPV-based transaction at a merchant's premises with a stolen card for which the adversary has previously obtained the relevant PIN.

In this work, we briefly described an OPV process that is based on publicly available information. The architecture of the payment network for the OPV and subsequently for the online transaction authorization was explained. This payment network and its associated deployment open up a potential route for an adversary to compromise it to his benefit. We detailed our assumptions regarding the payment network's operating environment, the capabilities of an adversary and potential attack scenarios. Subsequently, we proposed three potential ways to enhance the OPV process and a proposal of how to bind it to the online transaction authorization. Proposed solutions were then analyzed with a discussion on their security in the context of the adversary's capabilities.

It also provided the execution measurements for our proposed modifications; this showed the potential performance penalty incurred by our proposals. Furthermore, proposed modifications were then subjected to the mechanical formal analysis using the Casper FDR tool. The concerns raised by this work are considered to be valid as the OPV and online transaction authorization is considered the highest level of trust in the card-based payment mechanism. It can differ based on laws/regulations or the relationship between the cardholder and CIB, but if the correct PIN is used in an OPV and online transaction authorization then the liability of the payment is either with the cardholder or the CIB. If attacks can successfully occur at

this level they could potentially cause substantial reputation damage to the overall card-based payment scheme, along with causing financial loss to the cardholder/CIB. Furthermore, such attacks could make it difficult to detect whether an OPV based transaction was actually made by the cardholder or the adversary, as the compromise of the intermediary nodes might not be detected in time. Therefore, we consider this to be a concern and suggest that a mandating rollout of an OPV process in a geographical region should take into consideration these concerns and our potential solutions.

1.3 ALGORITHM

1.3.1 SVM CLASSIFICATION

Support Vector Machines (SVMs) are a popular classification algorithm used in credit card fraud detection. SVMs work by finding the best hyperplane that separates the data into different classes. In the context of credit card fraud detection, the classes are fraudulent and legitimate transactions. The SVM classification process starts with data preparation, where transaction data is cleaned, normalized, and scaled to ensure that each feature contributes equally to the model. Next, the feature extraction module identifies relevant features in the data and creates new features to improve the model's accuracy. After feature extraction, the data is split into training and testing sets. The training set is used to train the SVM model to learn patterns in the data and make accurate predictions. The SVM model then uses the training data to find the hyperplane that best separates the data into different classes. The hyperplane is chosen to maximize the margin between the classes, which is the distance between the hyperplane and the closest data points from each class. Once the SVM model has been trained, it can be used to predict whether a new transaction is fraudulent or legitimate. The model takes in the features of the transaction and outputs a prediction, which is

either a binary classification of fraudulent or legitimate. To evaluate the SVM model's accuracy, the testing data is used, and the model is fine-tuned to achieve better accuracy. Overall, SVMs are powerful machine learning algorithms that can be used to detect fraudulent transactions accurately. SVMs work by finding the hyperplane that best separates the data into different classes and can be fine-tuned to achieve high accuracy in detecting fraudulent transactions.

CHAPTER 2

LITERATURE SURVEY

2.1 "CREDIT CARD FRAUD DETECTION WITH A NEURAL-NETWORK" S. Ghosh, and D.L. Reilly et.al(5) 1994 has proposed The payment card industry has grown rapidly the last few years. Companies and institutions move parts of their business, or the entire business, towards online services providing e-commerce, information and communication services for the purpose of allowing their customers better efficiency and accessibility. Regardless of location, consumers can make the same purchases as they previously did "over the desk". The evolution is a big step forward for the efficiency, accessibility and profitability point of view but it also has some drawbacks. The evolution is accompanied with a greater vulnerability to threats. The problem with making business through the Internet lies in the fact that neither the card nor the cardholder needs to be present at the point-of-sale. It is therefore impossible for the merchant to check whether the customer is the genuine cardholder or not. Payment card fraud has become a serious problem throughout the world.

2.2 "CARDWATCH: A NEURAL NETWORK BASED DATABASE MINING SYSTEM FOR CREDIT CARD FRAUD DETECTION" E. Aleskerov, B. Freisleben, and B. Rao et.al(6) 1997 has proposed In this work, CARDWATCH, a database mining system used for credit card fraud detection, is presented. The system is based on a neural network Earning module, provides an interface to a variety of commercial databases and has a comfortable graphical user interface. Test results obtained for synthetically generated credit card data and an auto associative neural network model show very successful fraud detection rates. In this work, a neural network-based

database mining system for credit card fraud detection was presented. The system is easily extensible and able to work directly on a large variety of commercial databases. The current version of the system was tested on synthetically generated data using an auto associate with very promising results: a fraud detection rate of 85% and a legal transaction identification rate of 100% were achieved.

2.3 "A NOVEL AND SUCCESSFUL CREDIT CARD FRAUD DETECTION SYSTEM IMPLEMENTED IN A TURKISH BANK"Ekrem Duman, Ayse Buyukkaya, and Ilker Elikucuk et.al(12) 1999 has proposed and developed a credit card fraud detection solution for a major bank in Turkey. It had a great impact in the rule-based fraud detection process used by the bank. Indeed, while eighty percent of the rules have been eliminated and the number of alerts has been reduced to half, a significant increase in fraud detection has been recorded. As of now the system can catch ninety seven percent of fraud attempts online or, nearly online. The study is interesting in both the formulation of the problem and the algorithms implemented. In fact, we noticed that the standard classification algorithms are not fully suitable for the fraud detection problem (as the cost of every individual false negative can be different from the others), and we looked for alternative methods, especially the meta-heuristics.

2.4 "AGENT-BASED DISTRIBUTED LEARNING APPLIED TO FRAUD DETECTION" S. Stolfo and A.L. Prodromidis et.al(9), 1999 has proposed Inductive learning an classification techniques have been applied in many problems in diverse areas. In this work we describe an AI-based approach that combines inductive learning algorithms and meta-learning methods as a means to compute accurate classification models for detecting electronic fraud. Inductive learning algorithms are used to compute detectors

of anomalous or errant behavior over inherently distributed data sets and meta-learning methods integrate their collective knowledge into higher level classification models or meta-classifiers. By supporting the exchange of models or classifier agents among data sites, our approach facilitates the cooperation between financial organizations and provides unified and cross-institution protection mechanisms against fraudulent transactions. Through experiments performed on actual credit card transaction data supplied by two different financial institutions, we evaluate this approach and we demonstrate its utility.

2.5 "A COMPREHENSIVE SURVEY OF DATA MINING-BASED FRAUD DETECTION RESEARCH", C. **Phua, V.Lee, K. Smith, and R. Gayler et.al(8) 2006,** has proposed the term fraud here refers to the abuse of a profit organization's system without necessarily leading to direct legal consequences. In a competitive environment, fraud can become a business-critical problem if it is very prevalent and if the prevention procedures are not fail-safe. Fraud detection, being part of the overall fraud control, automates and helps reduce the manual parts of a screening/checking process. This area has become one of the most established industry/government data mining applications. It is impossible to be absolutely certain about the legitimacy of and intention behind an application or transaction. Given the reality, the best cost-effective option is to tease out possible evidences of fraud from the available data using mathematical algorithms Evolved from numerous research communities, especially those from developed countries, the analytical engine within these solutions and software are driven by artificial immune systems, artificial intelligence, auditing, database.

2.6 “RESEARCH ON CREDIT CARD FRAUD DETECTION MODEL BASED ON DISTANCE SUM”Wen-Fang Yu & Na Wang, et.al(4) 2011 has proposed Using graphs as to extracting and presenting data has a wide range of applications. Such applications may appear in detecting semantic and structural patters and exploiting graphs toward such applications have steadily been growing. In this work we are going to display one of the most perilous abnormalities in credit cards industry on such concept basis. With advancing technology in field of banking, the rate of use of credit cards has remarkably been escalated. Correspondingly frauds frequency has increased in this area which to surmount such anomalies we model them by means of graphs. Of the prominent advantage of proposed approach is drop of system overload rate during running computations in order to detecting frauds and consequently acceleration of detection speed.

2.7 "UNDERSTANDING CREDIT CARD FRAUDS"Tej Paul Bhatla, Vikram Prabhu, & Amit Dua et.al(2) 2011 has proposed Credit Card Fraud is one of the biggest threats to business establishments today. However, to combat the fraud effectively, it is important to first understand the mechanisms of executing a fraud. Credit card fraudsters employ a large number of modus operandi to commit fraud. In simple terms, Credit Card Fraud is defined as: When an individual uses another individuals’ credit card for personal reasons while the owner of the card and the card issuer are not aware of the fact that the card is being used. Further, the individual using the card has no connection with the cardholder or issuer, and has no intention of either contacting the owner of the card or making repayments for the purchases made. Increasingly, the card not present scenario, such as shopping on the internet poses a greater threat as the merchant (the web site) is no longer protected with advantages of physical verification such as signature check, photo identification, etc.

2.8 CREDIT CARD FRAUD STATISTICS,Statistic Brain et.al(3) 2014

Statistic Brain Research Institute has proposed and initiated the discussion by describing the current situation of the credit card business with respect to fraud issues. Although new technology is available and widely supported by banks and merchants globally to lessen or perhaps eradicate the repercussions of credit card fraud, some researchers are starting to challenge its design and implementation. This work suggests building a model based on the spending behavior of the card holders and using it to detect anomalous transactions. This work did not elaborate the details of the created model due to a non-disclosure agreement (NDA) between the participating bank and the proponent. However, this work was able to showcase the techniques and processes utilized in building the model. The proponent hopes that in the near future, this work will be used as a reference by some banks or individuals to implement fraud detection system in the financial sector. Benefits of implementing such detection system will lessen the phone and SMS costs shouldered by the banks; instead of sending SMS transaction notifications to all customers, message will be sent to those customers with detected anomalous transaction.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In this existing work the Naïve Bayes machine learning classifier tries to predict a class which is known as outcome class based on probabilities, and also conditional probabilities of its occurrence from the training data. This kind of learning is very efficient, fast and high in accuracy for real-world scenarios, and also this learning type is known as supervised learning.

The implementation of Naïve Bayes and oneR algorithm on same credit card dataset so as to calculate the precision of algorithms to identify the fraudulent transactions in the dataset. Experimental results depict that both classifiers work differently for the same dataset.

The purpose is to enhance the precision, accuracy and increase the flexibility of the algorithm. Bayesian network classifiers are very popular in the area of machine learning and it comes under the category of supervised classification models. Naïve Bayes classifier is also a well-known Bayesian Network that is based on Bayes theorem of conditional probability and hence, is a classifier based on probability which considers Naïve i.e., strong independence assumption.

It was formerly introduced with some other name, into the text retrieval community as a baseline technique for categorizing text because there was a problem of deciding in which category the documents do belongs to, with word frequencies as the feature.

The Naïve Bayes machine learning classifier tries to predict a class which is known as outcome class based on probabilities, and also conditional probabilities of how many times it occurred from the training data.

This kind of learning is very efficient, fast and high in accuracy for real-world scenarios, and is known as supervised learning. Also, this is highly efficient because it estimates the parameters by using very small training data which is used for classification and is based upon word independence.

Though Naïve Bayes is quite simple to implement and understand and uses strong assumptions. It gives pretty accurate results and also it has been proven over and over the time that Naïve Bayes works effectively in various areas related to machine learning

3.1.1 DRAWBACKS

- ✓ Less accuracy.
- ✓ Poor Prediction Result
- ✓ Cannot support dynamic Datasets.
- ✓ Lack in time performance.
- ✓ Data Analysis and Data Cleaning is a very hard job to do.
- ✓ The algorithm takes a lot of time to iterate over the large data set.
- ✓ Optimizing an individual algorithm for doing the whole task is hard.

3.2 PROPOSED SYSTEM

A proposed system for credit card fraud detection using Support Vector Machines (SVMs) could involve several key steps. First, transaction data would need to be collected from credit card companies or financial institutions, and then pre-processed by removing any unnecessary columns and encoding categorical features. Next, the data would be split into training and testing sets, and an SVM model would be trained on the training data using a suitable kernel function (such as linear, polynomial, or radial basis

function). The hyperparameters of the SVM model, such as the regularization parameter C and kernel function parameters, would be tuned to optimize its performance on the testing set. Once the model has been trained, it can be deployed to detect fraud in real-time credit card transactions. The model would analyse each transaction and predict whether it is fraudulent or not based on the learned patterns in the training data. The performance of the model would be evaluated using metrics such as accuracy, precision, recall, and F1-score. Comparisons could also be made between different models based on the use of SVMs or other machine learning algorithms to determine the most effective approach for detecting fraud.

3.2.1 ADVANTAGES

- SVMs are known to be robust to outliers in the data, which can be especially useful for credit card fraud detection where fraudulent transactions may be outliers in the data.
- SVMs can model non-linear decision boundaries, allowing them to capture complex relationships in the data that may be difficult for other algorithms to capture.
- SVMs use a subset of training data to define the decision boundary, which makes them more memory-efficient compared to other machine learning algorithms that require storing all the training data.
- SVMs can use different kernel functions, such as linear, polynomial, or radial basis functions, which can be tuned to optimize the model's performance for a specific problem.
- SVMs can be effective with small training sets, which can be beneficial for credit card fraud detection where fraudulent transactions are relatively rare.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS:

- Processor Type : Pentium i3
- Speed : 3.40GHZ
- RAM : 4GB DD2 RAM
- Hard disk : 500 GB
- Keyboard : 101/102 Standard Keys
- Mouse : Optical Mouse

4.2 SOFTWARE REQUIREMENTS:

- Operating System : Windows 10
- Front End : Jupyter Notebook/ Anaconda tool
- Coding Language : Python

4.3 SOFTWARE DESCRIPTION

4.3.1 ANACONDA

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, and anaconda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using. Cloud hosts hundreds of useful Python packages, notebooks, projects and environments for a wide variety of applications. You do not need to log in, or even to have a Cloud account, to search for public packages, download and install them.

You can build new anaconda packages using anaconda-build, then upload the packages to Cloud to quickly share with others or access yourself from anywhere. The Anaconda Cloud command line interface (CLI), anaconda-client, allows you to manage your account - including authentication, tokens, upload, download, remove and search. Connect to and manage your Anaconda Cloud account. Upload packages you have created. Generate access tokens to allow access to private packages.

For developers, Cloud is designed to make software development, release and maintenance easy by providing broad package management support. Cloud allows for free public package hosting, as well as package channels, providing a flexible and scalable service for groups and organizations of all sizes.

APPLICATIONS PROVIDED IN ANACONDA DISTRIBUTION

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

- Jupyter Lab
- Jupyter Notebook
- Qt Console
- Spyder
- Glue viz
- Orange3
- RStudio
- Visual Studio Code

Jupyter Lab: This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Jupyter Notebook: This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.

Qt Console: It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.

Spyder: Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.

VS Code: It is a streamlined code editor with support for development operations like debugging, task running and version control.

Glue viz: This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.

Orange 3: It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.

RStudio: It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks

4.3.2 JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and

much more. The software requirement specification is created at the end of the analysis task. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a representation of system behaviour, an indication of performance requirements and design constraints, appropriate validation criteria.

FEATURES OF JUPYTER NOTEBOOK

- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.
- The ability to execute code from the browser, with the results of computations attached to the code which generated them.
- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc.
- For example, publication-quality figures rendered by the matplotlib library, can be included inline.
    - In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.
    - The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.

CHAPTER 5

SYSTEM DESIGN

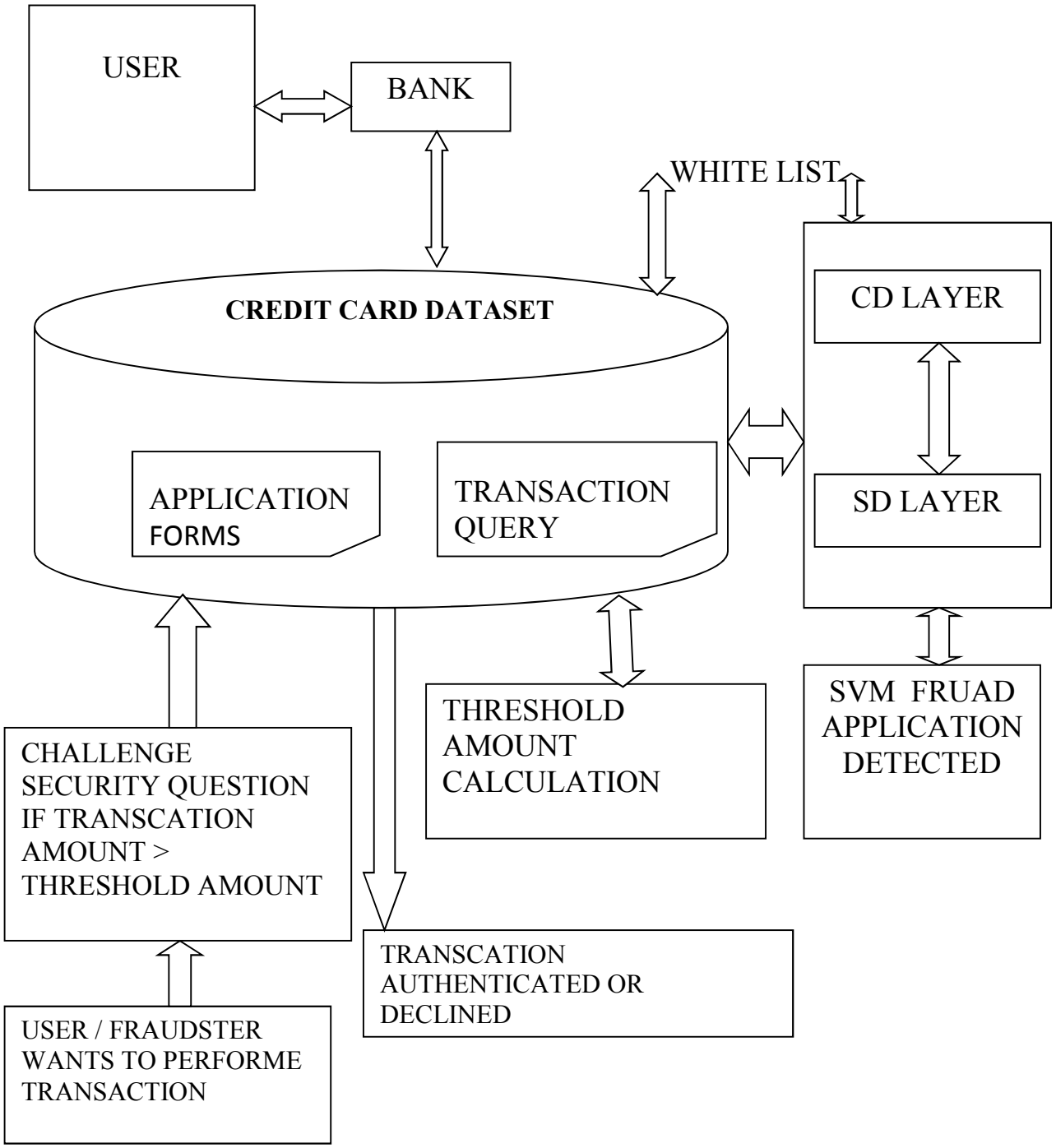

Fig:5.1 System Design

5.1 MODULE DESCRIPTION

5.1.1 LOAD INPUT DATA

Loading input data is a critical step in any credit card fraud detection system using SVMs. The transaction data must be collected from various sources, such as credit card companies or financial institutions, and then processed and prepared for use in the machine learning model. Here's a possible paragraph describing how input data could be loaded. The credit card fraud detection system would begin by loading the transaction data from various sources into the system. This data would typically include information such as the transaction amount, merchant category code, location, and time. The raw data would then need to be processed and cleaned to remove any duplicates, missing values, or other anomalies that could affect the accuracy of the model. Categorical features such as merchant category codes would need to be encoded using techniques such as one-hot encoding, while numerical features would need to be scaled to ensure that each feature contributes equally to the SVM model. Once the data has been preprocessed and prepared, it would be split into training and testing sets to enable the SVM model to learn patterns in the data and make accurate predictions. Overall, loading input data is a critical step in the credit card fraud detection process, and ensuring the data is clean and well-prepared is essential to achieving accurate results.

5.1.2 DATA PRE-PROCESSING

Data preprocessing is the process of cleaning, transforming, and organizing raw data before it is used for analysis or modeling. It involves a series of steps that are performed to ensure that the data is accurate, complete, consistent, and relevant. This involves removing or correcting any errors or inconsistencies in the data, such as missing values, duplicates, or incorrect

values. This involves converting the data into a suitable format for analysis or modeling. This may include scaling, normalization, or encoding categorical variables. This involves reducing the amount of data to be analyzed by selecting relevant features or samples, or by summarizing the data through techniques such as clustering or principal component analysis. This involves combining data from multiple sources into a single dataset for analysis.

5.1.3 FEATURE EXTRACTION

The feature extraction module is a crucial component of any credit card fraud detection system using SVMs. The module is responsible for identifying and creating new features from the raw transaction data that can be used to improve the SVM model's accuracy in detecting fraudulent transactions. Here's a possible paragraph describing how the feature extraction module could work. The feature extraction module would begin by identifying relevant features in the transaction data that could be used to distinguish between fraudulent and legitimate transactions. For example, the module could look for patterns in the transaction amount or frequency of transactions, or it could examine features such as the time of day or geographic location of transactions. Once relevant features have been identified, the module would create new features by combining and transforming the existing ones. This could involve aggregating transaction amounts over time to create features such as daily or weekly spending averages, or generating features based on the merchant category code, such as the proportion of transactions made at high-risk merchants. Other feature extraction techniques could include principal component analysis (PCA) or linear discriminant analysis (LDA) to reduce the dimensionality of the data and focus on the most important features. Overall, the feature extraction module is critical in improving the accuracy of the SVM model by creating new and informative features that capture the patterns and relationships in the transaction data.

CHAPTER 6

SYSTEM METHODOLOGY

6.1 Data Preparation

The dataset came from two (2) different sources: the fraud cases and the transaction log file. The former contains all the reported credit fraud incidents while the latter contains all the transactions accumulated by the participating bank. Moreover, data pertaining to the fraud cases were collected and encoded in a user defined worksheet while the transaction logs attributes have some similarities to ISO 8583 standard. most point-of-sale (POS) devices and card issuers conform to ISO 8583 – a standard for financial transaction card originated messages. Some attributes of the ISO 8583 standard are present in the transaction logs, although most card systems and credit card companies affix few important details to the log-file. Data annotation will commence by comparing the recorded fraud cases to the transaction logs. When a match is detected, the record in transaction log will have a class value of "true", otherwise "false".

Here, a new multi-layered detection system to combat identity crime in real-time, complemented with two additional layers: Communal Detection (CD) and Spike Detection (SD).

6.1.1. Communal Detection

Communal relationships are social or family relationships such as parent-child, brothers, sisters etc. The usability of communal detection can be explained with an example. Suppose there are two credit card applications with same address, landline phone number, date of birth; one with name as John Smith and other Joan Smith.;

This application can be interpreted in three ways: -

1. It is a fraudster attempting to obtain multiple credit cards using near duplicated data.

2. Possibly they are twins living in the same house and both are applying for a credit card.

3. Or it can be the same person applying twice, and has done a typing mistake.

The communal detection algorithm layer detects frauds from communal relationships. To account for legal behavior and data errors, CD is the whitelist-oriented approach on a fixed set of attributes. The whitelist, a list of communal and self-relationships between applications, is crucial because it reduces the scores of these legal behavior and false positives [1]. A false positive is an error in some evaluation process in which a condition tested for is mistakenly found to have been detected.

6.1.2. Spike Detection

SD layer complements CD layer. It strengthens CD by providing attribute weights which reflects the degree of importance of attribute (like name, phone number).

a) Data set for real application

Data Set Substantial identity crime can be found in private and commercial databases containing information collected about customers, employees, suppliers, and rule violators. The same situation occurs in public and government regulated databases such as birth, death, patient and disease registries; taxpayers, residents' address, bankruptcy, and criminals lists. To reduce identity crime, the most important textual identity attributes such as

personal name, Social Security Number (SSN), Date-of-Birth (DOB), and address must be used.

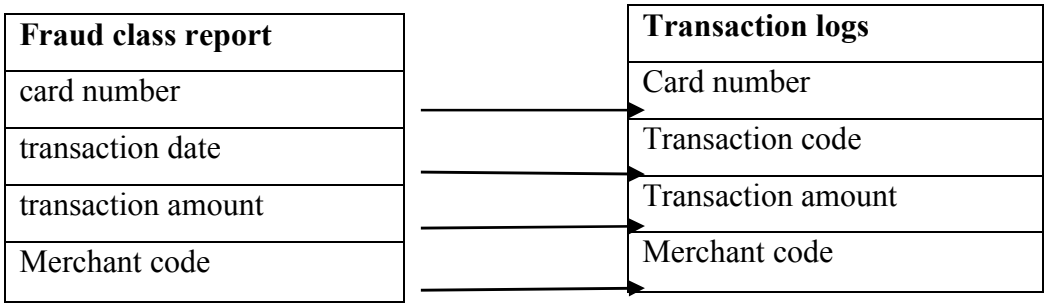

Fig:6.1.1 Data Mapping for Data Annotation

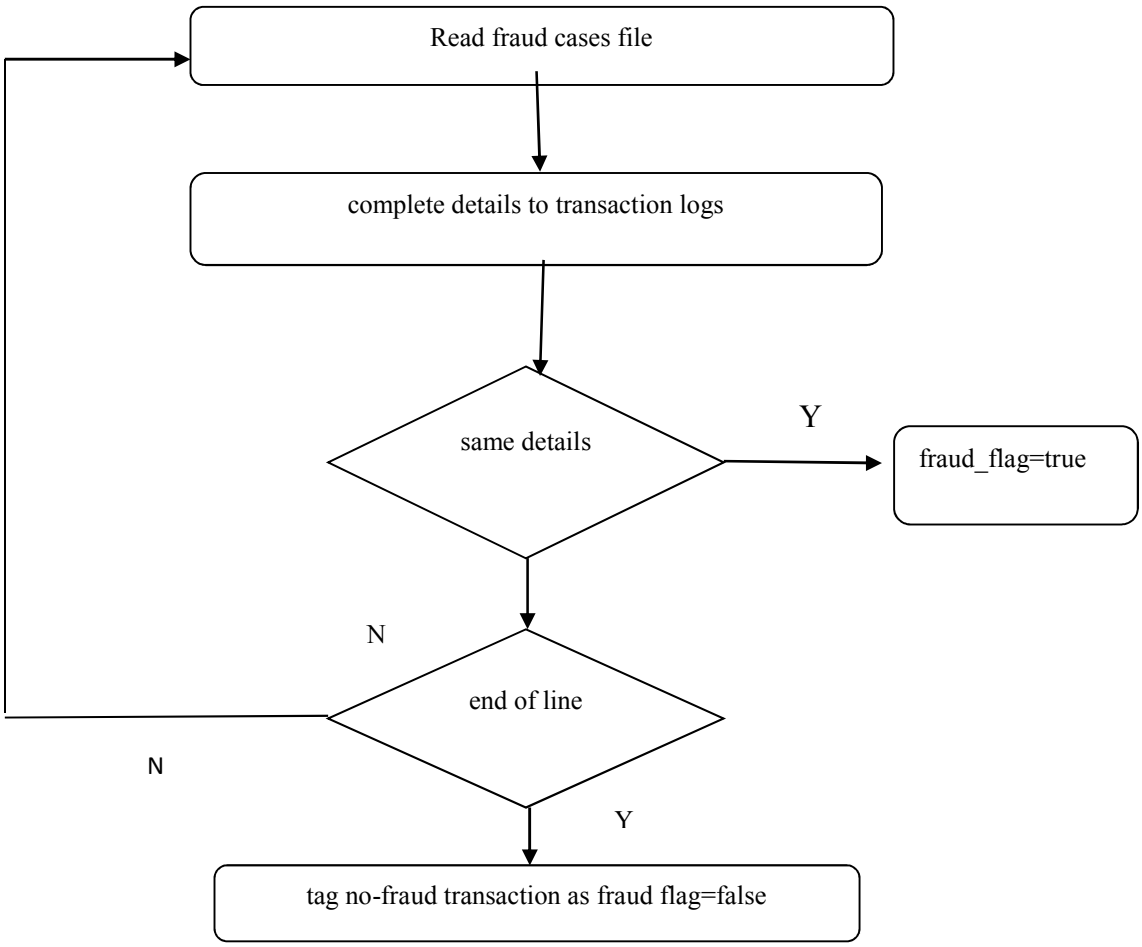

Fig:6.1.2. Data Annotation Process

6.2 Data Analysis

At the time of this writing, the participating bank had several million of transactions and several thousands of reported fraud cases; however, this work only utilized data from January to June of 2016. Evaluating the conditions of these dataset, it was obviously much skewed – the numbers of legitimate transactions versus fraudulent instances were way too far. According to the work, the preferred distribution must be 50:50 to be able to produce the ideal model. With this considered, and to eliminate the issue of data imbalance, a new dataset was created to contain legitimate transactions based from the card/account included in the recorded fraud case report. The newly generated dataset has a total count of 9,992 records; 9,733 of which are legitimate and 259 are fraudulent. Although this setup comprises of 3% fraudulent and 97% legitimate transactions, later, this dataset will be manipulated to formulate new datasets with different class distribution – this will confirm the effectiveness of the classifiers under evaluation. For the meantime, the dataset for the model creation and evaluation stage will use a 25% fraud and 75% normal transaction type concoction. Data in the transaction logs underwent several preprocessing tasks such as data-sanitation, normalization, binning, and handling null values. Prior to these activities, few attributes were removed from the dataset such as:

- account number, card number - this will eliminate the possibility of having a customer centric model.
- fields pertaining to dates - this will reduce the possibility of building a model focusing on date related events.
- And, control number pertaining to reported disputes - this will eliminate the possibility of creating a model directly referencing to this control number – since this field represents a potential fraud instance.

6.3 SYSTEM TESTING

System testing makes a logical assumption that, if all parts of the systems are correct, system testing is its utility as a user-oriented vehicle before implementation. The best program is worthless if it does not meet user needs. System testing identifies the errors, presenting the proposal to the administrator and changes the modification and also checks the reliability of output. Before implementation, the system is tested whether the required software and hardware are available to run this project.

This project has undergone the following testing procedures to ensure its correctness.

- **Unit testing**
- **Integration testing**
- **Validation testing**

6.3.1 Unit Testing

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules totes the interfaces before modules. Unit testing has proven its value in that a large percentage of defects are identified during its use.

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the form level testing is made. For example, storage of data to the table is in the correct manner. In this system, each form is considered as a separate unit and tested for errors. Every user input is unit tested for a valid accepted range.

6.3.2 Integration Testing

Integration testing sometimes called integration and Testing, abbreviated" I&T" is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrates system ready for system testing.

The purpose of integration testing is to verify functional performance, and reliability requirements placed on major design items. These "design items", i.e. assemblages (or groups of units), are exercised through their interfaces using Black box testing, success and error cases being simulated via appropriate parameter and data inputs.

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

6.3.3 Validation testing

Validation can be defined in many ways, but a simple definition is that can be reasonable expected by the clients, which is defined in the software requirement specification, a document that describes all user visible attribute of the software.

6.4 SYSTEM MAINTENANCE

The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In this project the process can be added without affecting other parts of the system.

Maintenance plays a vital role. The system is liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variations in its working environment. It has been seen that there are always some errors found in the system that must be noted and corrected. It also means the review of the system from time to time.

The review of the system is done for:

- Knowing the full capabilities of the system.
- Knowing the required changes or the additional requirements.
- Studying the performance.

CHAPTER 7

IMPLEMENTATION AND RESULT

7.1 CODING

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd
for dirname, _, filenames in os.walk('../smvc_credit/input'):
for filename in filenames:
print(os.path.join(dirname, filename))
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
 nunique = df.nunique()
 df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For
displaying purposes, pick columns that have between 1 and 50 unique values
 nRow, nCol = df.shape
 columnNames = list(df)
 nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi =
80, facecolor = 'w', edgecolor = 'k')
for i in range(min(nCol, nGraphShown)):
plt.subplot(nGraphRow, nGraphPerRow, i + 1)
 columnDf = df.iloc[:, i]
 if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
valueCounts = columnDf.value_counts()
```

```
valueCounts.plot.bar()
else:
columnDf.hist()
plt.ylabel('counts')
plt.xticks(rotation = 90)
plt.title(f'{columnNames[i]} (column {i})')
plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
plt.show()
# Correlation matrix
def plotCorrelationMatrix(df, graphWidth):
filename = df.dataframeName
df = df.dropna('columns') # drop columns with NaN
df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there
are more than 1 unique values
if df.shape[1] < 2:
print(f'No correlation plots shown: The number of non-NaN or constant
columns ({df.shape[1]}) is less than 2')
return
corr = df.corr()
plt.figure(num=None, figsize=(graphWidth, graphWidth), dpi=80,
facecolor='w', edgecolor='k')
corrMat = plt.matshow(corr, fignum = 1)
plt.xticks(range(len(corr.columns)), corr.columns, rotation=90)
plt.yticks(range(len(corr.columns)), corr.columns)
plt.gca().xaxis.tick_bottom()
plt.colorbar(corrMat)
plt.title(f'Correlation Matrix for {filename}', fontsize=15)
plt.show()
```

```
# Scatter and density plots
def plotScatterMatrix(df, plotSize, textSize):
df = df.select_dtypes(include =[np.number]) # keep only numerical columns
# Remove rows and columns that would lead to df being singular
df = df.dropna('columns')
df = df[[col for col in df if df[col].nunique() > 1]] # keep columns where there
are more than 1 unique values
columnNames = list(df)
if len(columnNames) > 10: # reduce the number of columns for matrix
inversion of kernel density plots
columnNames = columnNames[:10]
df = df[columnNames]
ax = pd.plotting.scatter_matrix(df, alpha=0.75, figsize=[plotSize, plotSize],
diagonal='kde')
corrs = df.corr().values
for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes
fraction', ha='center', va='center', size=textSize)
plt.suptitle('Scatter and Density Plot')
plt.show()
nRowsRead = 1000 # specify 'None' if want to read whole file
# creditcard.csv may have more rows in reality, but we are only
loading/previewing the first 1000 rows
df1 = pd.read_csv('../smvc_credit/input/creditcard.csv', delimiter=',', nrows =
nRowsRead)
df1.dataframeName = 'creditcard.csv'
nRow, nCol = df1.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
df1.head(5)
```

Correlation matrix:

```
plotCorrelationMatrix(df1, 8)
```

Scatter and density plots:

```
plotScatterMatrix(df1, 20, 10)
```

Split the data for training

```
y = df1['Class']
x = df1.drop(columns=['Time', 'Amount'])
x = x.to_numpy()
y = y.to_numpy()
y[np.where(y == 0)] = -1
w = np.ones(len(x[0]))
```

Implementation

Loss function

```
def hinge_loss_fn(y, y_hat):
return np.maximum(0,(np.ones(len(y)) - y * y_hat))
```

Forward function

```
def forward(x, w):
return np.dot(x, w)
```

SVM function

```
def svm_fn(x, y, w, epoch):
if (y * forward(x, w)) < 1:
return ( (x * y) + (-2  * (1/epoch) * w) )
else:
return (-2 * (1/epoch) * w)
```

Optimize function (SGD)

```
def optimize(w, dw, lr):
w += lr * dw
return w
```

Training function

```
def svm(x, y, w, epochs=1601, lr=0.001):
losses = []
for epoch in range(1,epochs):
for i, _ in enumerate(x):
dw = svm_fn(x[i], y[i], w, epoch)
w = optimize(w, dw, lr)
if(epoch % 100 == 0):
loss = hinge_loss_fn(y, np.dot(x, w)).mean()
losses.append(loss)
print("Epoch ", epoch, " - Hinge Loss ", loss)
return w, losses
w = np.zeros(len(x[0]))
w, losses = svm(x, y, w)
pred = forward(x, w)
```

Results analysis

```
plt.title('SVM training')
plt.xlabel('# of epochs by 200')
plt.ylabel('# of loss percentage')
plt.plot(losses)
def accuracy(y_hat, y):
# get the number of card frauds predicted
pred_fraud = len(np.where(np.ceil(pred) > 0)[0])
# actual fraud numbers
actual_fraud = len(np.where(y > 0)[0])
return pred_fraud/actual_fraud
accuracy(pred, y)
```

7.2 RESULT

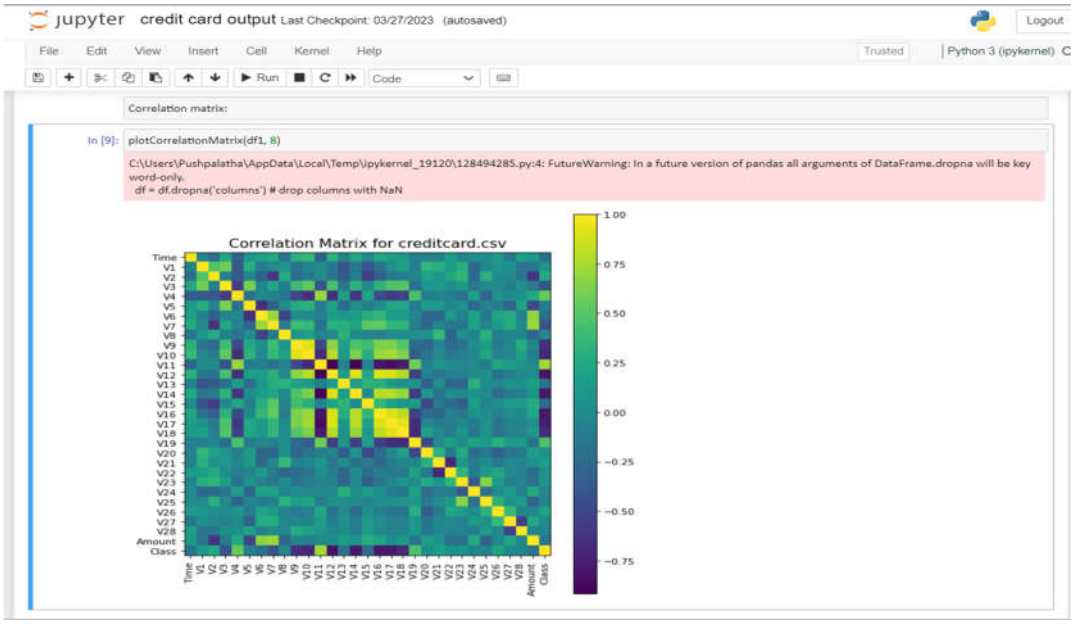

Fig:7.2.1 Correlation Matrix

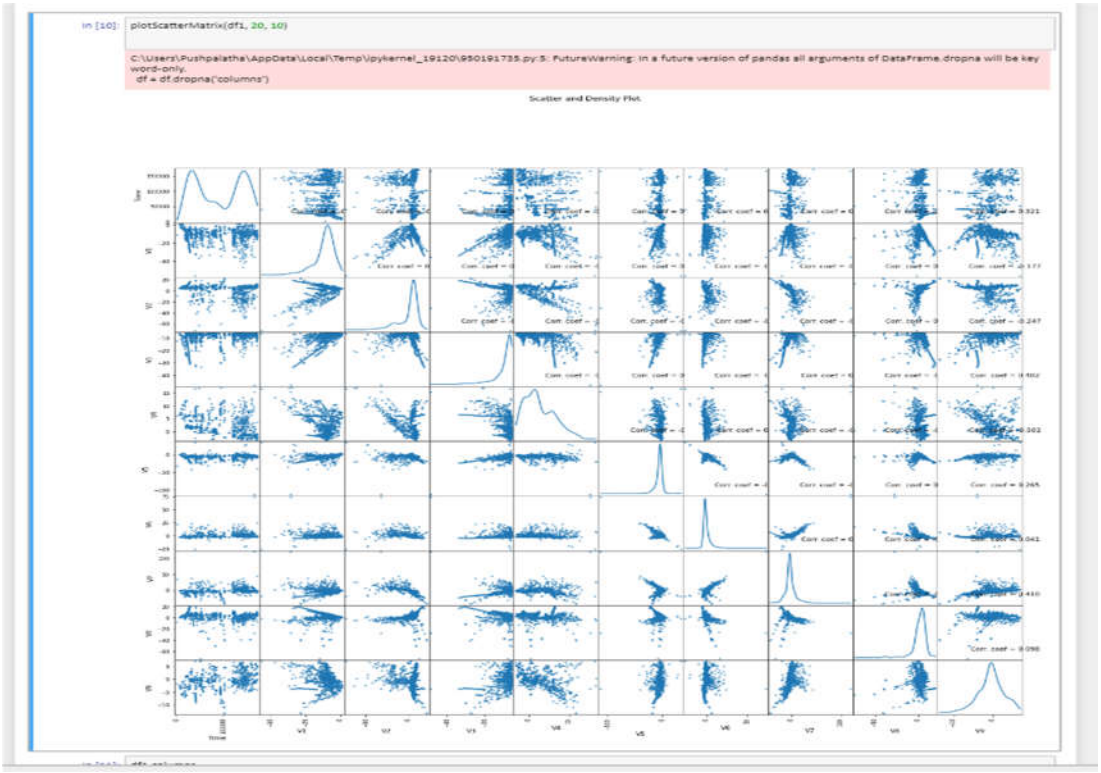

Fig:7.2.2 Scatter and Density Plot

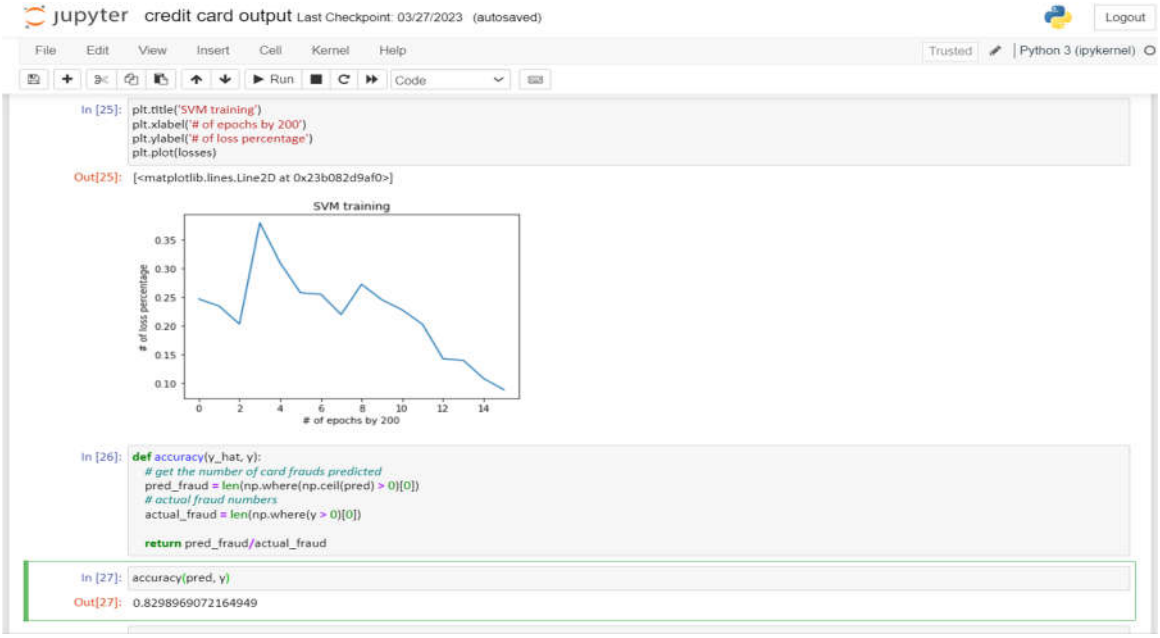

Fig:7.2.3 Accuracy

7.3 PERFORMANCE EVAULATION

The percentage of correctly classified instances (%CCL), ROC area or AUC and F-measure for all the six classifiers along with different degrees of imbalance for Euro Credit data and German Credit data sets.

The classifier with the highest AUC in each of the distribution is marked bold for each dataset. Analyzing keenly about the statistics we got from the classifiers, it is revealed that the two datasets performed differently for the same classifiers even having the same distributions.

Firstly the results we got for the Australian data sets were found to be with more correctly classifies instance, high ROC area and F-measure. Among the six classifiers with increase in imbalance ratio the classifiers tend to behave almost equally.

With increased imbalance ratio the classifiers LMT and RF gave equivalent results for the Australian data while for the German Credit data the results for SVM and RF were comparable. The results can be easily visualized from the ROC curves for both the datasets.

7.4 PERFORMANCE METRICS

1. **Precision value:** Preciseness benefit specific for the restored document. That is determined by way of the quantity of applicable datasets lost by way of the entire number of resultant datasets.

$$Precision\ value = \frac{True\ positive}{(True\ positive + False\ positive)}$$

2. **Recall value:** Recall value is specified to as the relevant datasets that are related to the other request Search.

$$Recall\ value = \frac{True\ positive}{(False\ positive + False\ Negative)}$$

3. **F measure:** The F measure is the harmonic mean of precision and recall

$$F\ measure = 2 * \frac{precision * recall}{precision + recall}$$

4. **Accuracy:** Accuracy gives the required related datasets used for classification. Calculate the proportion of true positive and true negative in all evaluated cases.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

Figure .4.1. The particular evaluation performance metrics of try to remember cost along with the suggested approach improved SVM classifier obtained try to remember cost is 93% plus progress as compared with present methods. It

also illustrates evaluation performance metrics from the Correct Beneficial Pace cost along with the suggested approach improved SVM classifier obtained 71 % plus progress as compared with present methods. The particular evaluation performance metrics of exactness along with the suggested approach improved SVM classifier obtained exactness cost is 93 % plus progress as compared with present methods. The particular suggested approach improved SVM classifier obtained accuracy cost is 68 %. Provide better result in comparison with other existing algorithms.

comparison with other existing algorithms.

Algorithm Used	TP Rate	FP Rate	Precision	Recall
ANN	0.771	0.266	0.788	0.746
Logistic Regression	0.845	0.155	0.846	0.849
OneR	0.855	0.131	0.866	0.856
SVM	**0.846**	**0.157**	**0.846**	**0.849**

Table:7.4.1. Analysis that comparison between Existing System with SVM (Proposed System) with parameter evaluation

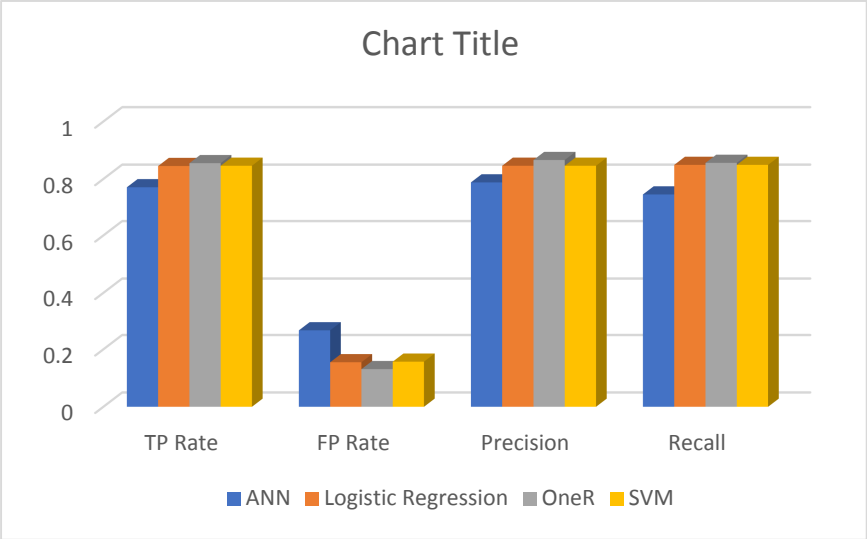

Fig:7.4.2. Graphical Parameters Comparison for Existing work with Proposed SVM algorithm

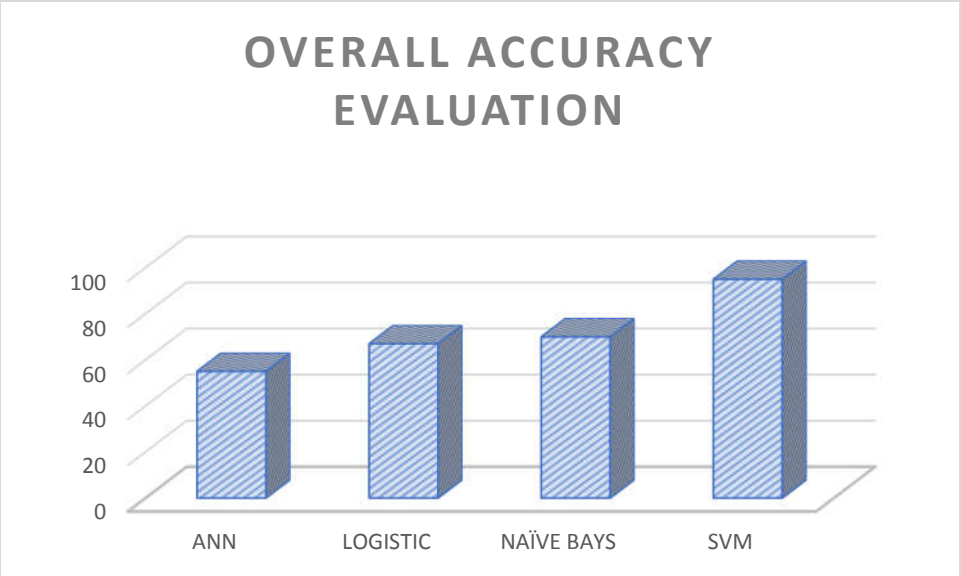

Fig:7.4.3 Analyzed for F1 Score (F-Measure) and Overall accuracy compared Existing work with SVM.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In conclusion, credit card fraud detection using SVMs is an effective way to combat the growing problem of fraudulent transactions. SVMs are powerful machine learning algorithms that can learn patterns in the data and make accurate predictions about new transactions. By identifying relevant features in the transaction data and creating new features to improve the accuracy of the model, SVMs can detect fraudulent transactions with high accuracy. The SVM classification process involves data preparation, feature extraction, model training, and prediction, all of which work together to detect fraudulent transactions. The SVM model's accuracy is evaluated using testing data, and the model can be fine-tuned to achieve better accuracy. Credit card fraud is a serious problem that affects both individuals and financial institutions. With the increasing prevalence of credit card fraud, it is essential to use sophisticated tools like SVMs to detect fraudulent transactions accurately. By using SVMs for credit card fraud detection, financial institutions can protect their customers' assets and prevent financial losses due to fraudulent activities.

8.2 FUTURE ENHANCEMENT

To further improve credit card fraud detection using SVMs, future research can explore several areas. One potential avenue for enhancement is to incorporate more advanced techniques for feature extraction, such as deep learning algorithms like convolutional neural networks (CNNs) or recurrent neural networks (RNNs). These techniques can automatically learn relevant

features from raw data and may improve the accuracy of the model. Another possible area of improvement is to use ensemble methods such as random forests or gradient boosting to combine multiple SVM models and improve the overall accuracy. These techniques can also help to address the issue of imbalanced data sets, where fraudulent transactions are rare compared to legitimate transactions. Additionally, integrating real-time monitoring and alerts for suspicious transactions can further improve fraud prevention. This would involve continuously monitoring transaction data and using predictive models to identify potentially fraudulent transactions in real-time, triggering alerts to fraud investigators or blocking transactions if necessary. Overall, these future enhancements can help to make credit card fraud detection using SVMs even more accurate and efficient, ultimately helping to protect financial institutions and their customers from fraudulent activities.

BIBLOGRAPHY

1. Steven J. Murdoch, Saar Drimer, Ross Anderson, and Mike Bond, "Chip and PIN is Broken" in IEEE Symposium on Security and Privacy, 2021
2. Tej Paul Bhatla, VikramPrabhu, & Amit Dua, "Understanding Credit Card Frauds".Tata Consultancy Services.
3. Statistic Brain Research Institute (2014, July 12). Credit Card Fraud Statistics (2021).
4. Wen-Fang Yu & Na Wang, "Research on Credit Card Fraud Detection Model Based on Distance Sum" in International Joint Conference on Artificial Intelligence,
5. S. Ghosh, and D.L. Reilly, "Credit Card Fraud Detection with a Neural-Network", Proc. 27th Hawaii International Conference on System Sciences: Information Systems: Decision Support and Knowledge-Based Systems, vol. 3, pp. 621-630, 2021.
6. E. Aleskerov, B. Freisleben, and B. Rao, "CARDWATCH: A Neural Network Based Database Mining System for Credit Card Fraud Detection", Proc. IEEE/IAFE: Computational Intelligence for Financial Engineering, pp. 220-226, 2020.
7. R. Brause, T. Langsdorf, and M. Hepp, "Neural Data Mining for Credit Card Fraud Detection," Proc. IEEE Int'l Conf. Tools with Artificial Intelligence, pp. 103-106,
8. C. Phua, V.Lee, K. Smith, and R. Gayler, "A Comprehensive Survey of Data Mining-Based Fraud Detection
9. S. Stolfo and A.L. Prodromidis, "Agent-Based Distributed Learning Applied to Fraud Detection," Technical Report CUCS-014-99, Columbia Univ., 2021.

10. C. Phua, D. Alahakoon, and V. Lee, “Minority Report in Fraud Detection: Classification of Skewed Data,” ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 50-59, 2021.
11. Varun Chandola, Arindam Banerjee, & Vipin Kumar, "Anomaly Detection: A Survey" in ACM Computing Surveys, 2021 ©, Vol. 41, No. 3, Article 15, pp 15:1 - 15:58
12. Philip K Chan, Wei Fan, Andreas Prodomidis, Salvatore Stolfo, "Distributed Data Mining in Credit Card Fraud Detection". Copyright 2020
13. Ekrem Duman, Ayse Buyukkaya, and Ilker Elikucuk, "A Novel and Successful Credit Card Fraud Detection System Implemented in a Turkish Bank" in 2013 IEEE 13th International Conference on Data Mining Workshops, © 2013. DOI 10.1109/ICDMW.2021.168.
14. W. Fang. YU, and N. Wang, “Research on Credit Card Fraud Detection Model Based on Distance Sum”, International Joint Conference on Artificial Intelligence, 2020.
15. C.B. Necib, J. Freytag, “Ontology based query processing in database management systems”, Proceeding on the 6th international on ODBASE, pp. 37-99, 2021.