Asian College of Higher Studies

Tribhuvan University

Institute of Science and Technology

A Project Report

ON

Credit Card Fraud Detection

Submitted to

Institute of Science and Technology

Department of Computer Science and Information Technology

Tribhuvan University

In partial fulfilment of the requirements for the **Bachelor's Degree in Computer Science and Information Technology (BSC.CSIT)**

Submitted By

Amul Shrestha (7762/072)

Ashim Rajbhandari (7764/072)

Binod Jung Bogati (7769/072)

Diwash Shrestha (7773/072)

under the supervision of

Mr. Tej Bahadur Shahi

DATE

Letter Of Approval

This is to certify that the project carried out by **Mr. Amul Shrestha**, **Mr. Ashim Rajbhandari**, **Mr. Binod Jung Bogati** and **Mr. Diwash Shrestha** entitled "**Credit Card Fraud Detection**" in partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Evaluation Committee

Mr. Tej Bahadur Shahi
Project Supervisor
Name of Department
Asian College of Higher Studies
Dhobidhara, Kathmandu
Nepal

Mr. Name of Department Chief
Head of Department
Name of Department
Name of your college
Address
Nepal

Mr. Name
Internal Examiner
Name of Department
Name of your college
Address
Nepal

Mr. Name
External Examiner
Name of Department
Name of College
Address
Nepal

Recommendation

It is certified that **Mr. Amul Shrestha (7762/072)**, **Mr. Ashim Rajbhandari (7764/072)**, **Mr. Binod Jung Bogati (7769/072)** and **Mr. Diwash Shrestha (7773/072)** has carried out the project work entitled "**Credit Card Fraud Detection**" under my supervision.

I recommend the project in the partial fulfilment for the requirement of **Bachelor's Degree of Science in Computer Science and Information Technology**.

.....

Mr. Tej Bahadur Shahi

Supervisor

Date: .....

Acknowledgements

First of all, we would like to show our deepest gratitude and appreciation to our college Asian College of Higher Studies for providing and accommodating a suitable environment to advance in our project work and complete it successfully.

Then, we would also want to show our appreciation to our supervisor, Mr. Prarup Gurung for administering and providing guidance and also for being an exemplary supervisor one can only hope for throughout our project. Without your assistance, it would have been a very challenging task for us to complete this project.

We want to thank our coordinator Mr. Brihat Boshwa for his continuous support and help throughout our project and for managing and accommodating every necessary resources and materials that was required in the project.

Lastly, we would like to specially thank and acknowledge our classmates who knowingly or unknowingly were a valuable part in every possible way even if it is in the smallest aspects of the contribution towards the completion of the project.

Abstract

Credit card fraud is theft and fraud committed using or involving a payment card i.e. credit card, as a fraudulent source of funds in a transaction. In current days, Credit Card is an easy way nowadays to use as a payment system for shopping in retail shops, online stores or be used to withdraw cash money from ATM Machines. Banks are also liberal to issue the credit cards these days than in older days. So, the more usage of the credit cards for withdrawing the money, the more chances of fraud cases. The change in technology and rush in human behavior has made the need for payment systems important in the daily life of the people. With the use of the payment system and withdrawal of money by credit card, the issue of security in the sense of physical and technical is also increased and that must be dealt to provide a better service. The purpose of this project is to develop a System which evaluates the transactions conducted through credit cards and produce an outcome in terms of 0 and 1. Here 1 is the term which represent fraud case or a probability of fraud and the term 0 represents the opposite i.e. not a fraud case. The data sets will be passed to the system in the set of chunks i.e. data will be fed to the system in parts and the system will predict if the transactions performed are fraud or not.

Keywords: *PCA transformation; XGBoost; GBM; Random Forest; logloss; AUCPR; Classification; R Programming; H2O; Shiny; GitHub.*

Table of Contents

Page

Acknowledgements **i**

Abstract **ii**

Table of Contents **iii**

List of Figures **vi**

List of Abbreviations **vii**

1 Introduction **1**

1.1 Background 1

1.2 Problem Statement 2

1.3 Objectives 2

1.4 Scope and Limitation 2

1.5 Project Road-map 3

1.6 Methodology 3

1.6.1 Input Transactional Data 4

1.6.2 Data Input to Model 4

1.6.3 Evaluation 4

1.6.4 Classification 4

1.6.5 Output 5

2 Literature Review **6**

2.1 Fraud Detection in Credit Card by Clustering Approach 6

2.2 Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy . 6

3 Requirement Analysis **7**

3.1 Project Requirements 7

3.1.1 Functional Requirements 7

3.1.2 Non-Functional Requirements 7

3.2 Feasibility Study 8

3.2.1 Technical Feasibility . . . 8
3.2.2 Economic Feasibility . . . 9
3.2.3 Operational Feasibility . . . 9
3.2.4 Schedule Feasibility . . . 9
3.3 Hardware and Software Requirement . . . 10
3.4 Used Tools . . . 10
3.4.1 R Programming Language . . . 10
3.4.2 H20 Machine Learning . . . 11
3.4.3 Shiny Web Framework . . . 11
3.4.4 R Studio . . . 11
3.4.5 Git/GitHub . . . 11
3.4.6 Lime Package . . . 12

4 System Design **13**
4.1 High Level Design . . . 13
4.1.1 Use Case Diagram . . . 13
4.1.2 Sequence Diagram . . . 14

5 System Implementation and Testing **16**
5.1 Implementation Methodology . . . 16
5.2 Algorithms . . . 16
5.2.1 Random Forest . . . 16
5.2.2 Gradient Boosting Machine . . . 17
5.2.3 XGBoost . . . 17
5.3 Analysis . . . 18
5.3.1 Random Forest . . . 18
5.3.2 GBM . . . 19
5.3.3 XGBoost . . . 20

6 Conclusion and future works **21**
6.1 Conclusion . . . 21

6.2 Future works 21

References **23**

Appendices **24**

List of Figures

1.1 Methodology 4

3.1 Gantt Chart 9

4.1 Use Case Diagram 14

4.2 Sequence Diagram 15

List of Abbreviations

CSS	Cascading Style Sheet
GBM	Gradient Boosting Machine
IDE	Integrated Development Environment
PCA	Principal Component Analysis
RFC	Random Forest Classification
SMOTE	Synthetic Minority Over-sampling Technique
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

1.1 Background

With the growing use of technology, banking fraud has also increased a lot. Among different banking fraud, credit card has been a target for fraudsters because it's a widespread and growing mode of payment.

Credit card fraud is theft and fraud committed using or involving a payment card i.e. credit card, as a fraudulent source of funds in a transaction. Credit card fraud happens when someone uses your credit card or credit account to make a purchase you didn't authorize. For example, Fraudsters can also steal the credit card account number, expiry date and other details to make unauthorized transactions, without needing your physical credit cards.

Our project, credit card fraud detection problem includes modeling of the past credit card transactions. This model is used to identify if a new transaction is fraudulent or not. Our goal is to detect fraudulent transactions and also minimizing the incorrect fraud classifications.

In our project, we will be using credit card transaction data sets we have acquired from www.kaggle.com. Data sets mainly consists of two-days transactions from European Cardholders and the variables of data sets are the result of a PCA transformation due to confidentiality issues. Before starting to build a model, we have gone through series of algorithms like XGBOOST, Gradient Boosting Machine (GBM), Random Forest and implemented our data set s to these algorithms to see the performance in terms of logloss, AUCPR, Confusion Matrix. After evaluating the performance in above terms, we will be choosing the best algorithm suited for the system. The final built model will be deployed in our system in web-based technology which will classify the new transaction data as fraudulent or genuine transaction.

The data sets we have chosen to process through our model is an imbalance data set because the data set contains maximum number of genuine transactions which outnumber the fraudulent transaction in great number.

1.2 Problem Statement

The credit cards are the common target for fraudsters because they can earn a lot of money without taking many risks. This is because crime is often only discovered a few weeks after date of incident and fraudsters can easily escape committing fraud.

Fraud detection in credit card is the process of identifying those transactions which are fraudulent. It is a classification problem of the credit card transactions with two classes of legitimate or fraudulent. This system tried to take the data from everyday transaction and process them to the system that evaluated the fraudulent transactions and warned the respective stakeholders to take appropriate actions.

1.3 Objectives

The principal objective of the project is to categorize everyday occurring credit card transactions in to fraudulent transactions and non-fraudulent transactions. Some of the objectives that this project aims to fulfill are:

- To develop a software which detects fraudulent transactions made by using credit cards.
- To provide reporting system for fraud investigators.

1.4 Scope and Limitation

The project is intended to deliver a web-based system for detecting credit card fraud. Due to limited knowledge, resources, and time, we were not able to deliver real-time system but we are planning to implement it as a future enhancement. Our system detects the fraud only after transactions is made. The target audience for this project would be fraud investigators.
This project is mainly directed to the solution provided for the financial organizations like Banking System where a lot of credit card issuance and transactions are frequent.

1.5 Project Road-map

Talking about the road-map of our project, initially we, the members of the project decided to gather to a common place to discuss about the plans and proceedings of the project. As we decided the topic of the project we were going to work-on, we also had a vision of activities we were going to process to have successful completion of this project. The different activities carried out during the completion the project were:

- Requirement Collection
- Design of System
- Implementation and Coding
- Testing and Maintenance
- Documentation

The details of above activities are describes properly in upcoming chapters where we can know about each phases of our project. And these Activities gives us clear idea of how all the proceedings of our work were done in each and every phases of the project. All the phases of our project were completed in sequential order one after another.

1.6 Methodology

The Credit Card Fraud Detection System runs through the following phases to detect the fraud and legal transactions.

- Input
- Transaction Data
- Data Input to Model
- Evaluation
- Classification
- Output

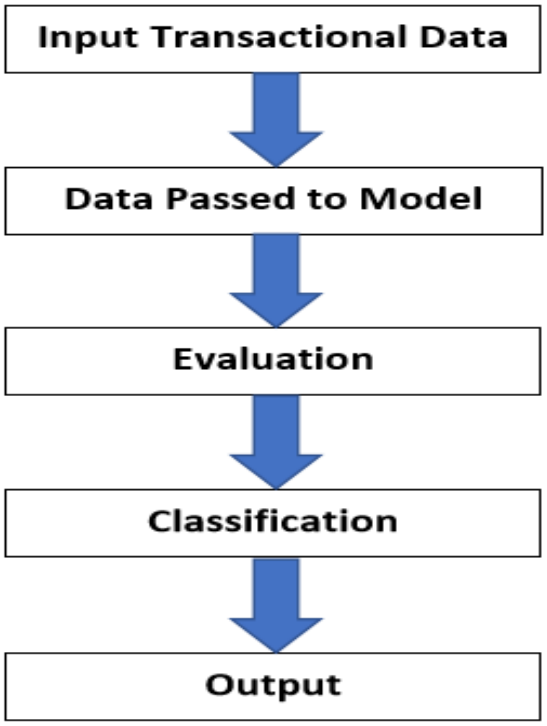


Figure 1.1: Methodology

1.6.1 Input Transactional Data

The transactional data set is divided into number of chunks. Each chunk is sent to the model separately to be evaluated by the Fraud Detection System.

1.6.2 Data Input to Model

The chunk data input from the user is passed to the machine learning model which computes each transaction in the data chunk and finds out whether the transaction is fraud or not.

1.6.3 Evaluation

The fraud detection system is trained to the data sets which consists of both fraud and not fraud data. When the system is ready, we send the chunk of new raw data for the purpose of evaluation.

1.6.4 Classification

As we are building a system based on the classification problem, the data are classified into number of sub-categorizations. Here in our systems, the machine learning model classifies the data set into

two categories – 0 and 1 where 0 represents for non-fraud transaction and 1 represents for fraud transaction.

1.6.5 Output

Lastly, the system provides the user with the classified data and the user can visualize output in the User Interface of the Web-Based System.

Chapter 2

Literature Review

2.1 Fraud Detection in Credit Card by Clustering Approach

*V*aishali, The author[1] states to make a better understanding of how we can detect fraudulent transactions or any outliers which may affect the organization with the help of clustering methods. Fraudulent transactions have been in an increasing manner which contradicts the accountability of an organization. So, this paper describes about the different possible methods through which we can find detects these anomalies or outliers that can be found in transactions by using Clustering approach and its different methods of clustering. Clustering is a process of arranging data into groups of similar objects. This paper mostly focuses on K-means clustering algorithm which is an unsupervised technique. They use .NET language on Visual studio 2012 to apply the algorithm in their data. Then they write a pseudo code of the new algorithm and then the clusters formed are then visualized into different range of risk.

2.2 Credit Card Fraud Detection: a Realistic Modeling and a Novel Learning Strategy

*A*ndrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi, The author[2] approaches the fraud detection in a different manner. They talk about different mechanisms regulating a real-world fraud detection system and then provide a formal model of the articulated classification problem to be addressed in the fraud detection. So, they first give details to the operating conditions of a real-world fraud detection system and then model the problem and present the most suitable performance measure. They describe the main peculiarities and the operating conditions of real-world fraud detection system. After that, they provide/propose an effective learning strategy for addressing the arising challenges, including the verification latency and alert feedback interaction. Then this strategy in used or tested over a large number of credit card transactions.

Chapter 3

Requirement Analysis

3.1 Project Requirements

The system requirement analysis is done in order to acquire the details of system and desired functionality of the system efficiently which includes both the hardware and software requirement for the satisfactory output and functionality of the application.
The requirement are divided into functional and non- functional requirement. In functional requirement the activities and services that system must provide were included which may be in terms of input, output and processing. In the non-functional requirement characterized the performance, accuracy schedule and flexibility.

3.1.1 Functional Requirements

Functional requirement specifies the system should provide. How the system behaves on particular input and how the system should behave to particular situation. In our system, '**Credit Card Fraud Detection System**', the functional requirements are as follows:

- System should be able to take input as the credit card transactional data.
- System should be able to process the input data.
- System should be able to categorize transactions as fraudulent or legal.
- System should be able to show the result on screen.

3.1.2 Non-Functional Requirements

Non- functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards.
Nonfunctional requirements often apply to the system as a whole.

- **Performance** - Accuracy of '**Credit Card Fraud Detection System**' should be higher than 90%.

- **Functionality** – Our system should be able to classify the input data correctly as per the above-mentioned functionalities.
- **Availability** – The system will be available to process data offline and online as per need of the user.
- **Learning Ability** – The system is designed in such a way that it is easy to understand the functionality and operations just by viewing the interface.
- **Reliability** – As per our system's accuracy level i.e. 90% or above, we can say the result given by the system is reliable.

3.2 Feasibility Study

A feasibility study is an analysis that takes all of a project's relevant factors into account— including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Feasibility studies discern the pros and cons of undertaking a project before lots of time and money are invested into it.

The goal of a feasibility study is to thoroughly understand all aspects of a project, concept, or plan; become aware of any potential problems that could occur while implementing the project; and determine if, after considering all significant factors, the project is viable—that is, worth undertaking.

There are four key considerations involved in the feasibility analysis are:

- **3.2.1 Technical Feasibility**

  The technical portion of the system consists of various tools and algorithms to design and develop the system "Credit Card Fraud Detection System" which are selected carefully and meet high standard for designing the System Architecture and developing the model. Various online tools like Lucid Chart and offline tools like R-Studio with associated packages are used to develop the system which confirms the reliability of the System.

- **3.2.2 Economic 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 in to the research and development of the system is limited. But it also necessary to expend the company budget which allows to not only to run the company smoothly but also minimize economic loss to the company. Credit Card Companies are hugely budgeted and profitable. So, it is not a huge issue for budget to spend for the project.

- **3.2.3 Operational Feasibility**

With highly interactive and easy to use feature of the model, it will not be an issue for learn to use the System. So, training cost should be very low and processing of the data be smooth and faster. Basic Knowledge of Computer System can be handy to use the system and understand the functionality.

- **3.2.4 Schedule Feasibility**

Our Project is scheduled as per the following Gantt Chart and is feasible to produce a result in scheduled time period.

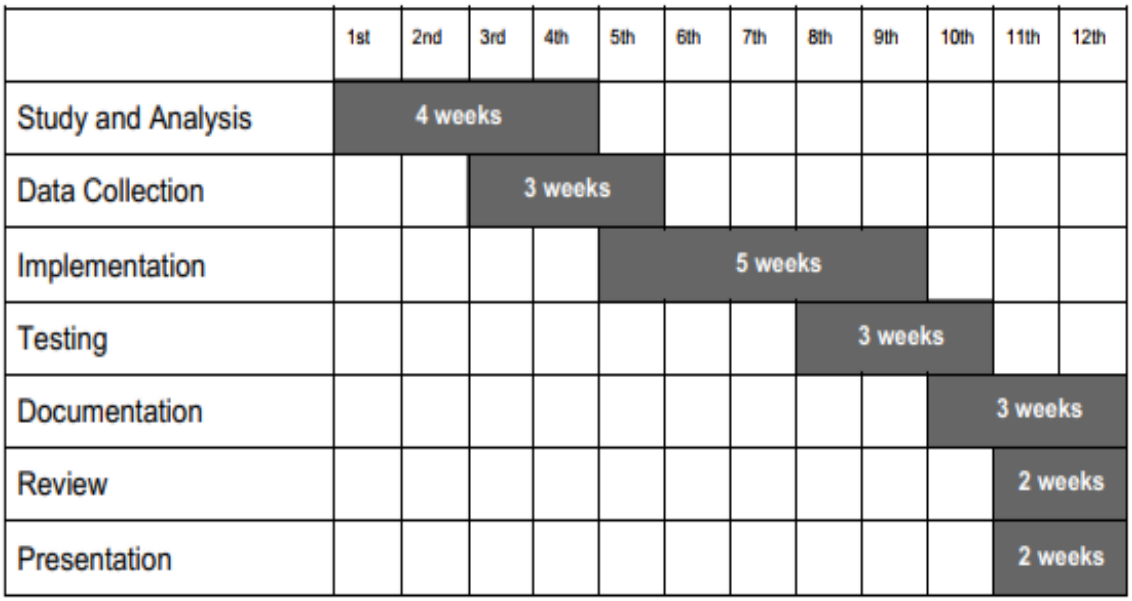


Figure 3.1: Gantt Chart

3.3 Hardware and Software Requirement

Laptop or Personal Computer

- System: Dual-Core Processor 2.4 GHz or above
- Operating System: Windows 7 or above
- Hard-Disk: 250 GB or Higher
- Monitor: 15 VGA or advance
- RAM: minimum 4GB or Higher

3.4 Used Tools

3.4.1 R Programming Language

R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, . . .) and graphical techniques, and is highly extensible.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility
- a suite of operators for calculations on arrays, in particular matrices
- a large, coherent, integrated collection of intermediate tools for data analysis
- graphical facilities for data analysis and display either on-screen or on hardcopy
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

In our project, R Programming has been used in data processing, visualization (Graphics), Machine Learning model building.

3.4.2 H20 Machine Learning

H2O is a fully open source, distributed in-memory machine learning platform with linear scalability. H2O supports the most widely used statistical & machine learning algorithms including gradient boosted machines, generalized linear models, deep learning and more. H2O also has an industry leading AutoML functionality that automatically runs through all the algorithms and their hyperparameters to produce a leader board of the best models. The H2O platform is used by over 18,000 organizations globally and is extremely popular in both the R & Python communities.

H2O is used in our project to create and train machine learning models based on XGBOOST, GBM, RandomForest Algorithms.

3.4.3 Shiny Web Framework

Shiny is an R package that makes it easy to build interactive web apps straight from R. We can host standalone apps on a webpage or embed them in R Markdown documents or build dashboards. We can also extend your Shiny apps with CSS themes, html widgets, and JavaScript actions.

Using Shiny, we will be creating dashboard where you can visualize transactional data and classify the input data using machine learning models.

3.4.4 R Studio

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).

R-Studio is used as an IDE for developing the entire project.

3.4.5 Git/GitHub

Git is a free and open source version controlling tool which is used for tracking changes in the computer files and coordinating work on those files among multiple people. Git was developed

by Linus Torvalds on 7 April, 2005 and can be used in almost every operating system including Windows, Linux, and Mac OS etc.

GitHub is a web-based hosting service for version control using git. It supports every feature of git and has also its own features. GitHub is used in web browsers like chrome, Firefox, edge etc.

3.4.6 Lime Package

Lime package is a visualization tool which helps to explain the individual predictions. Moreover, it's often important to understand the ML model that you've trained on a global scale, and also to zoom into local regions of your data or your predictions and derive local explanations. Global interpretations help us understand the inputs and their entire modeled relationship with the prediction target, but global interpretations can be highly approximate in some cases. Local interpretations help us understand model predictions for a single row of data or a group of similar rows.

Chapter 4

System Design

4.1 High Level Design

High Level Design explains the complete architecture of the software or system that is going to be developed. It tells about logical designs as well as physical implementation of the design. Architecture used to design the system gives us an overview of the entire system, identifying the main components that would be developed for the product and their interfaces. At initial phases of the design different components of the system that can be the part of the system are identified and in later part of the design, these components are brought together to fit-in in the system as a whole and work together to function a particular purpose.

The main focus sits at the Conceptual and Logical levels of abstraction for a project. And the stress should be all on diagrams and description of the system components.

4.1.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

During the development of Credit Card Fraud Detection System, we have used following diagrams for understanding the requirements and control structure of the system.

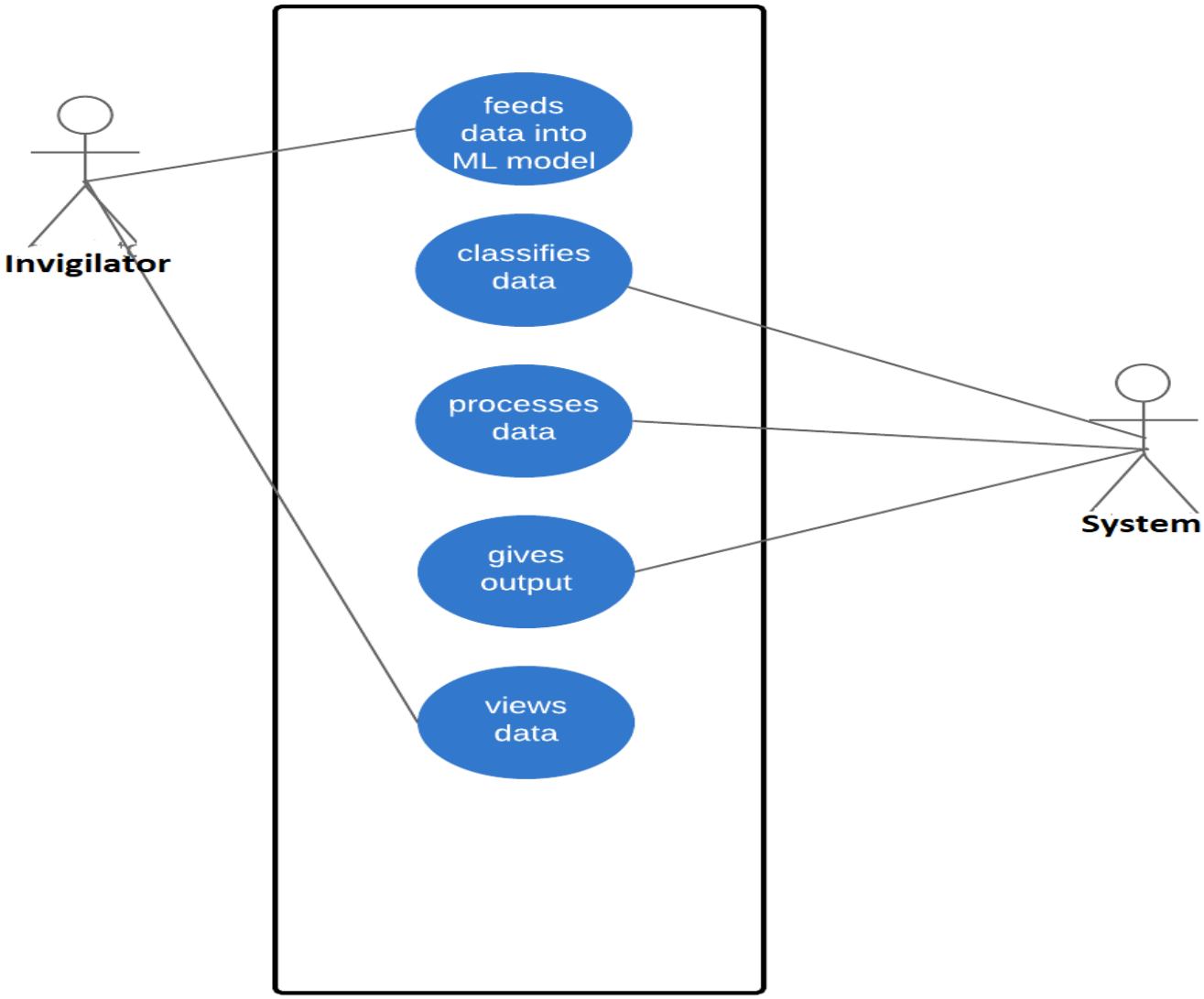


Figure 4.1: Use Case Diagram

4.1.2 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

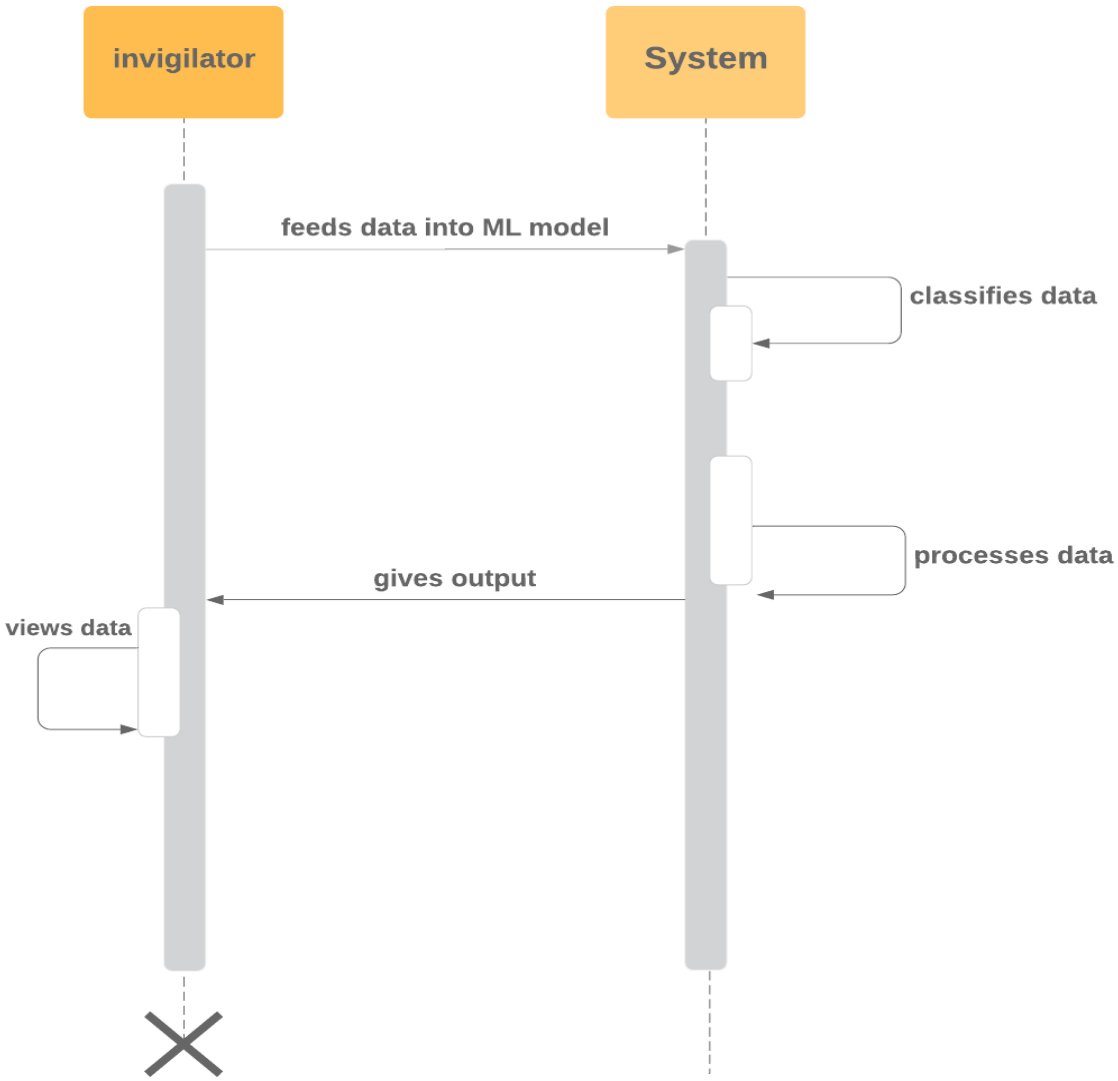


Figure 4.2: Sequence Diagram

Here in above Sequence Diagram, we can see the interaction between Invigilator and the System (Credit Card Fraud Detection System). Firstly, Invigilator feeds the data provided into Machine Learning model and the system captures data input such that it processes further to categorize data into fraudulent and non fraudulent data.

Chapter 5

System Implementation and Testing

5.1 Implementation Methodology

During the initial phase of the project, discussions about the selection of project topic and requirement collection as well as data set needed for the training of the machine learning system were gathered. Data collected for training and testing of the system were collected from www.kaggle.com. Due to the concern in privacy of customers, data found in www.kaggle.com comprised of limited parameters along with time and amount of transactions. After proper analysis on the collected requirements and data sets, designing of system using various diagrams were followed by its implementation. As different algorithms were used to implement the design of system, comparison of those algorithms in terms of performance and other different parameters were done to say which of them was better for our project. R Programming Language was used to develop the system along with implementation of different machine learning and interpretation algorithms.

5.2 Algorithms

5.2.1 Random Forest

Random forests is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Working Details of Random Forest

- Subset the original data so that the decision tree is built on only a sample of the original data-set.
- Subset the independent variables (features) too while building the decision tree.
- Build a decision tree based on the subset data where the subset of rows as well as columns is used as the data-set.
- Predict on the test or validation data-set.

- Repeat steps 1 through 3 n number of times, where n is the number of trees built.
- The final prediction on the test dataset is the average of predictions of all n trees.

5.2.2 Gradient Boosting Machine

Gradient refers to the error, or residual, obtained after building a model. Boosting refers to improving. The technique is known as gradient boosting machine, or GBM. Gradient boosting is a way to gradually improve (reduce) error.

In random forest the trees grown in parallel to get the average prediction across all trees, where each tree is built on a sample of original data. Gradient boosting, on the other hand, does the predictions using a different format.

Instead of parallelizing the tree building process, boosting takes a sequential approach to obtaining predictions. In gradient boosting, each decision tree predicts the error of the previous decision tree—thereby boosting (improving) the error (gradient).

It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

Working Mechanism of Gradient Boosting Machine

- Initialize predictions with a simple decision tree.
- Calculate residual - which is the (actual-prediction) value.
- Build another shallow decision tree that predicts residual based on all the independent values.
- Update the original prediction with the new prediction multiplied by learning rate.
- Repeat steps 2 through 4 for a certain number of iterations (the number of iterations will be the number of trees).

5.2.3 XGBoost

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve

many data science problems in a fast and accurate way.

5.3 Analysis

5.3.1 Random Forest

Validation:

MSE: 0.001510449

RMSE: 0.0388645

LogLoss: 0.01315081

Mean Per-Class Error: 0.09262784

AUC: 0.9456586

pr_auc: 0.8410997

Gini: 0.8913171

Training:

MSE: 0.03123014

RMSE: 0.1767205

LogLoss: 0.09278564

Mean Per-Class Error: 8.775471e-06

AUC: 0.9999955

pr_auc: 0.7805182

Gini: 0.999991

Testing:

MSE: 0.001356892

RMSE: 0.03683602

LogLoss: 0.01001509

Mean Per-Class Error: 0.07956311

AUC: 0.9576613

pr_auc: 0.8809002

Gini: 0.9153226

5.3.2 GBM

Training

H2OBinomialMetrics: gbm ** Reported on training data. **

MSE: 0.0002207475

RMSE: 0.01485757

LogLoss: 0.001552473

Mean Per-Class Error: 0.05744413

AUC: 0.9887471

pr_auc: 0.9061022

Gini: 0.9774941

Validation:

** Reported on validation data. **

MSE: 0.000425063

RMSE: 0.02061706

LogLoss: 0.003045673

Mean Per-Class Error: 0.1018695

AUC: 0.9745854

pr_auc: 0.8296786

Gini: 0.9491708

Testing:

MSE: 0.0003124859

RMSE: 0.01767727

LogLoss: 0.002096704

Mean Per-Class Error: 0.07391661

AUC: 0.9866222

pr_auc: 0.8501062

Gini: 0.9732443

5.3.3 XGBoost

Training:

** Reported on training data. **

MSE: 0.0001317273

RMSE: 0.01147725

LogLoss: 0.0006964352 Mean Per-Class Error: 0.03040833

AUC: 0.9999519

pr_auc: 0.9843941

Gini: 0.9999039

Validation:

** Reported on validation data. **

MSE: 0.0003952218

RMSE: 0.01988019

LogLoss: 0.002754937

Mean Per-Class Error: 0.0879894

AUC: 0.9809923

pr_auc: 0.8475315

Gini: 0.9619846

Training:

H2OBinomialMetrics: xgboost

MSE: 0.0002608651

RMSE: 0.01615132

LogLoss: 0.001772105

Mean Per-Class Error: 0.06821713

AUC: 0.9886354

pr_auc: 0.8794378

Gini: 0.9772707

Chapter 6

Conclusion and future works

6.1 Conclusion

"**Credit Card Fraud Detection System**" is an essential part of this today's financial industry to make flow of transaction secure and make valid payment scene. This system is essential to control illegal way of transaction of digital cash or conversion of digital cash to physical cash. Our project is based on the idea of keeping track of suspicious transaction that might happen to be an illegal commerce and report to the concerned authority to prevent such activities of transaction which is the possibility of fraud case.

The necessity of such system in the current scenario is very important as the number of financial organization issuing Credit Cards is increasing and with the increase of transactions made online which occurs in frequent manner, the case of fraud commerce also increase even in small quantity at numerous times or in huge quantity at once. So, this kind of system will help those organization to stop those criminals to thieve from others.

Here we have used R programming language and R-Studio IDLE for developing which gives a easy and convenient platform to work with codes and data and also linking packages essential for specific work is easy and efficient with good performance of the system. Various Machine Learning Algorithms have made the predictions to categorize genuine transactions and fraudulent proceedings efficient and reliable as the performance of those algorithms along with the interpretation of the fraudulent cases were nearly as needed. In the future, such algorithms which gets upgrades with time for better functioning may also help to catch the thief at the time of transaction or very soon after the dealings has been made.

6.2 Future works

The system we have developed to detect fraudulent transactions which has happened in the past and also, predict the fraudulent transactions that may occur in the future using various machine learning algorithms and interpretation algorithms. In future, the system we have developed may have few

changes to make it possible to detect such illegal transactions in real time such that possible actions can be taken immediately after the breach in the system.

References

[1] V. Vaishali, "Fraud detection in credit card by clustering approach," *International Journal of Computer Applications*, vol. 98, pp. 29–32, 07 2014.

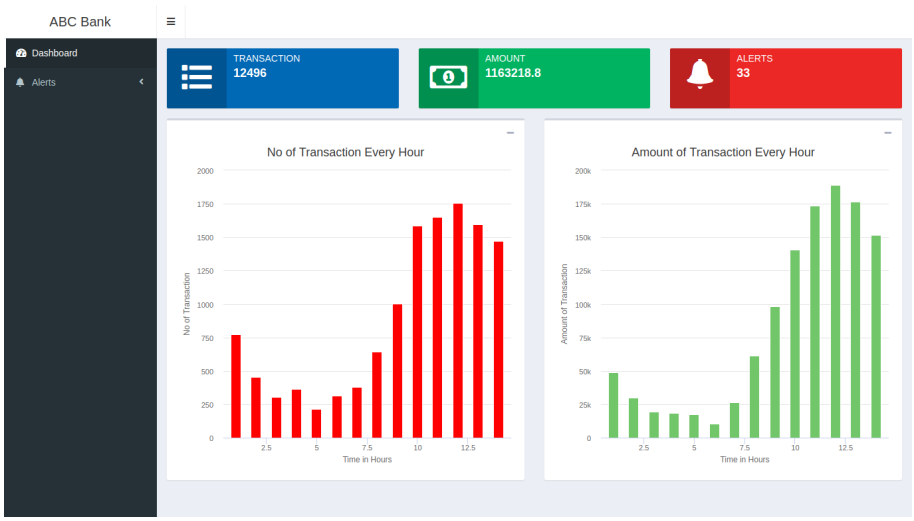
[2] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, Aug 2018.

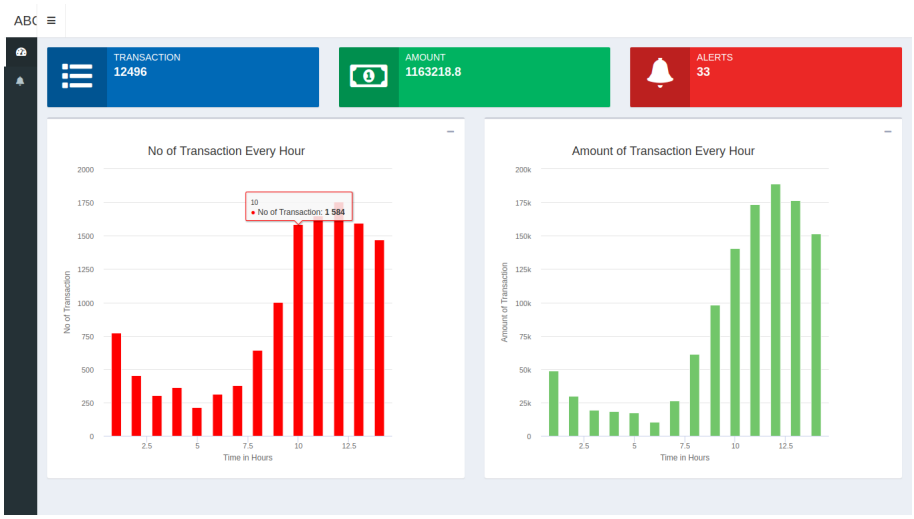
[3] M.-J. Lesot and A. Revault d'Allonnes, "Credit-card fraud profiling using a hybrid incremental clustering methodology," in *Scalable Uncertainty Management*, E. Hüllermeier, S. Link, T. Fober, and B. Seeger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 325–336.

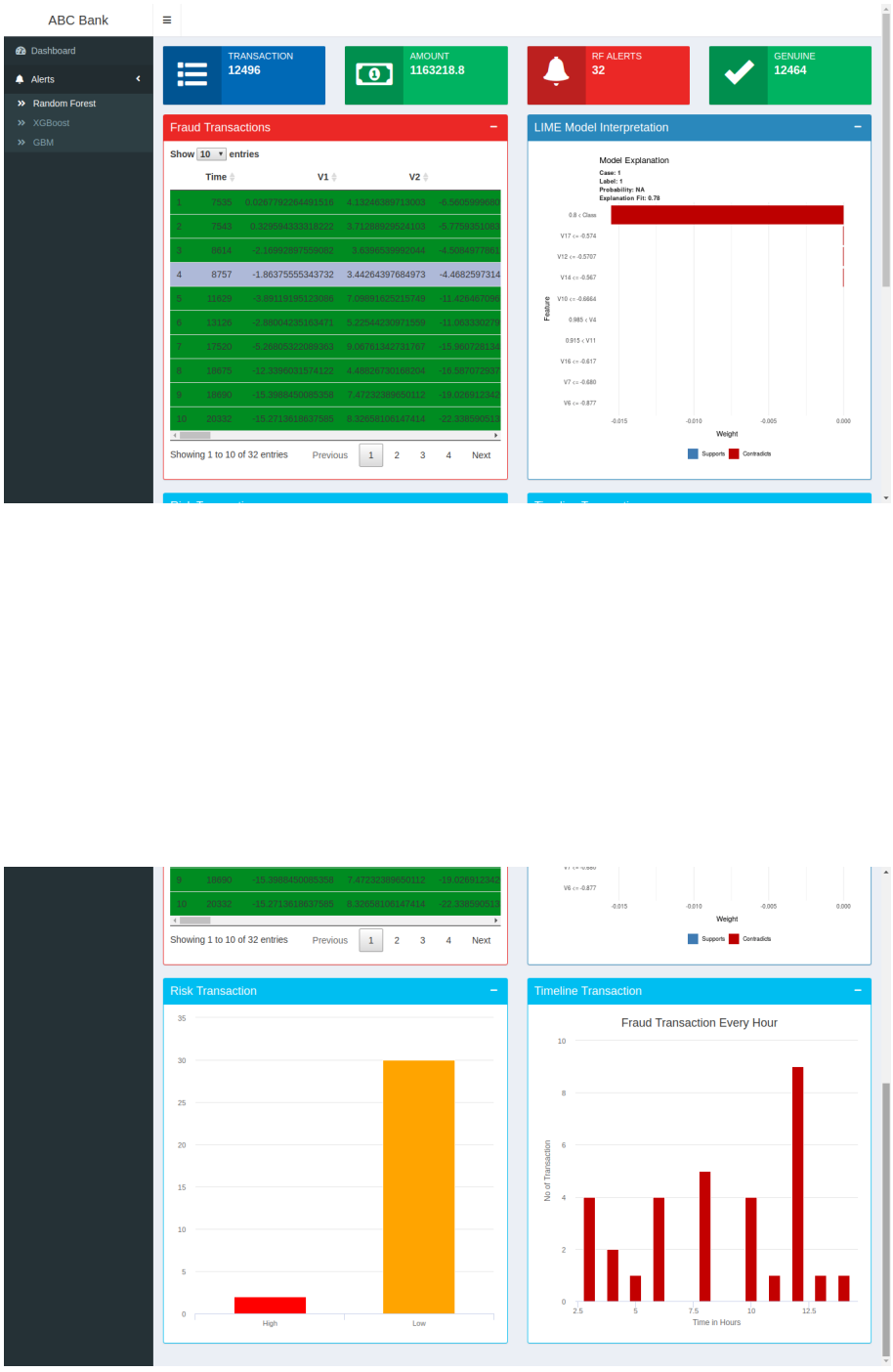
[4] A. D. Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, "Calibrating probability with undersampling for unbalanced classification," in *2015 IEEE Symposium Series on Computational Intelligence*, Dec 2015, pp. 159–166.

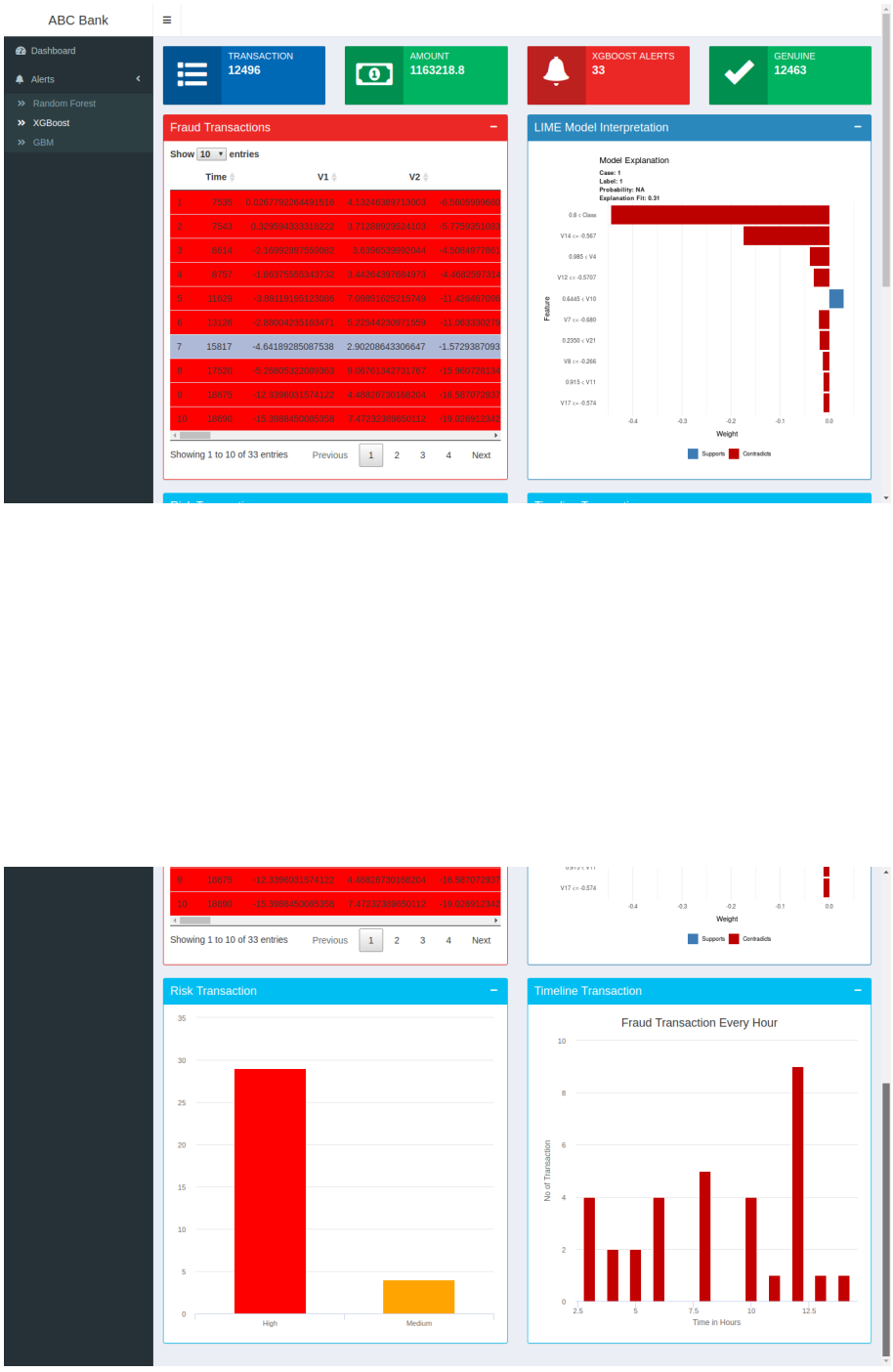
[5] M. R. Lepoivre, C. O. Avanzini, G. Bignon, L. F. Legendre, and A. K. Piwele, "Credit card fraud detection with unsupervised algorithms," 2016.

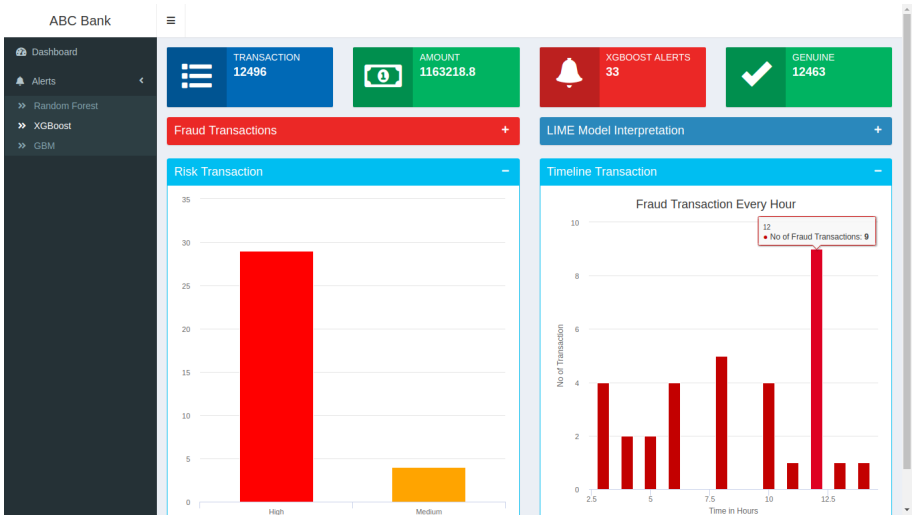
Appendices

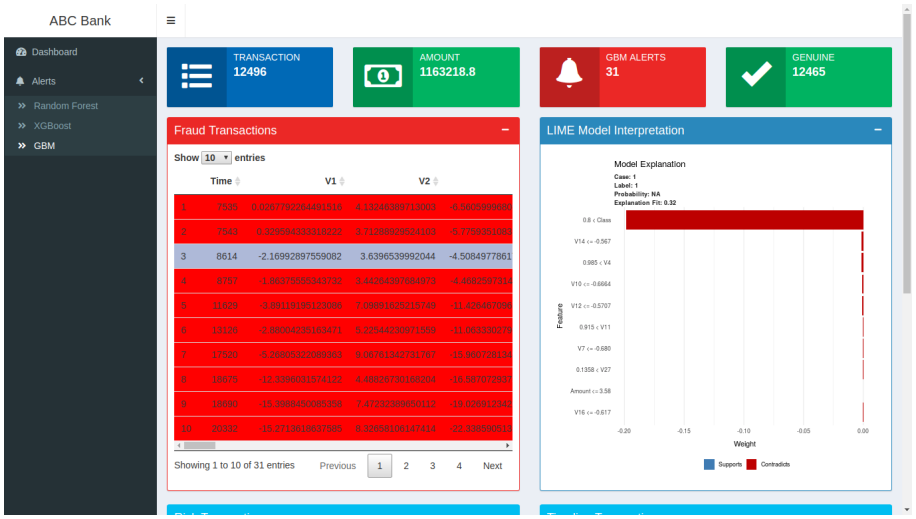

ABC Bank
Dashboard
Alerts
TRANSACTION
12496
AMOUNT
1163218.8
ALERTS
33
No of Transaction Every Hour
No of Transaction
Time in Hours
Amount of Transaction Every Hour
Amount of Transaction
Time in Hours

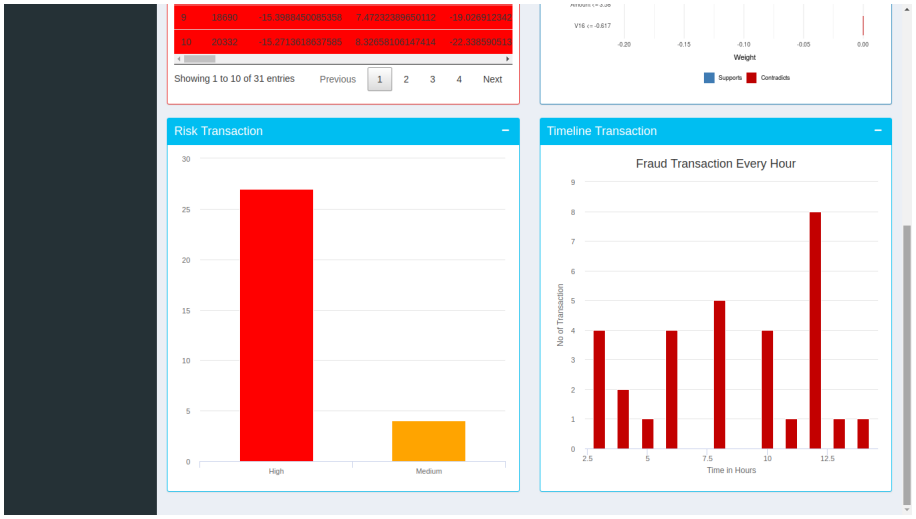

ABC
TRANSACTION
12496
AMOUNT
1163218.8
ALERTS
33
No of Transaction Every Hour
10
No of Transaction: 1 584
No of Transaction
Time in Hours
Amount of Transaction Every Hour
Amount of Transaction
Time in Hours

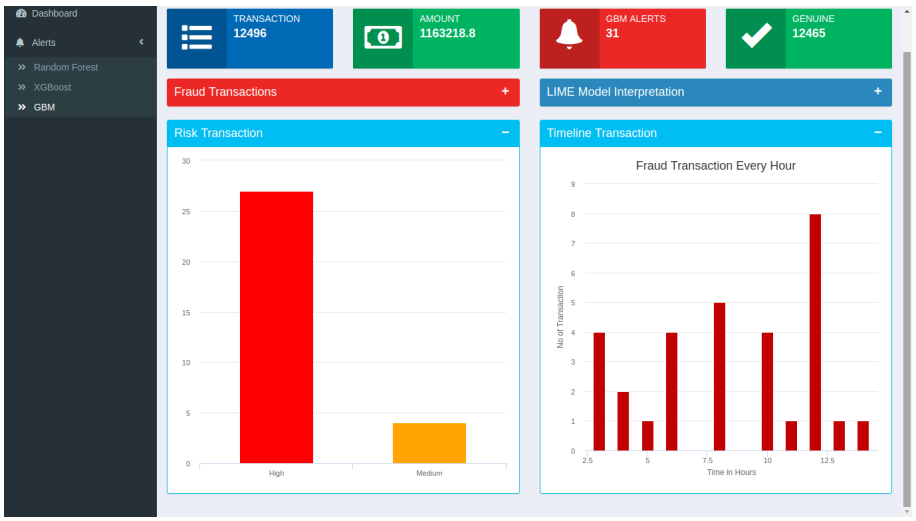

ABC Bank
Dashboard
Alerts
Random Forest
XGBoost
GBM
TRANSACTION
12496
AMOUNT
1163218.8
RF ALERTS
32
GENUINE
12464
Fraud Transactions
Show 10 entries
Time
V1
V2
Showing 1 to 10 of 32 entries
Previous
1
2
3
4
Next
LIME Model Interpretation
Model Explanation
Case: 1
Label: 1
Probability: NA
Explanation Fit: 0.78
Feature
Weight
Supports
Contradicts
Risk Transaction
High
Low
Timeline Transaction
Fraud Transaction Every Hour
No of Transaction
Time in Hours


ABC Bank
Dashboard
Alerts
Random Forest
XGBoost
GBM
TRANSACTION
12496
AMOUNT
1163218.8
XGBOOST ALERTS
33
GENUINE
12463
Fraud Transactions
Show 10 entries
Time
V1
V2
Showing 1 to 10 of 33 entries
Previous
1
2
3
4
Next
LIME Model Interpretation
Model Explanation
Case: 1
Label: 1
Probability: NA
Explanation Fit: 0.31
Feature
Weight
Supports
Contradicts
Risk Transaction
High
Medium
Timeline Transaction
Fraud Transaction Every Hour
No of Transaction
Time in Hours


ABC Bank
Dashboard
Alerts
Random Forest
XGBoost
GBM
TRANSACTION
12496
AMOUNT
1163218.8
XGBOOST ALERTS
33
GENUINE
12463
Fraud Transactions
LIME Model Interpretation
Risk Transaction
Timeline Transaction
Fraud Transaction Every Hour
No of Fraud Transactions: 9
No of Transaction
Time in Hours
High
Medium


ABC Bank
Dashboard
Alerts
Random Forest
XGBoost
GBM
TRANSACTION
12496
AMOUNT
1163218.8
GBM ALERTS
31
GENUINE
12465
Fraud Transactions
LIME Model Interpretation
Show 10 entries
Time
V1
V2
Showing 1 to 10 of 31 entries
Previous
Next
Model Explanation
Feature
Weight
Supports
Contradicts


Showing 1 to 10 of 31 entries
Previous
1
2
3
4
Next
Weight
Supports
Contradicts
Risk Transaction
Timeline Transaction
Fraud Transaction Every Hour
No of Transaction
Time in Hours
High
Medium


Dashboard
Alerts
Random Forest
XGBoost
GBM
TRANSACTION
12496
AMOUNT
1163218.8
GBM ALERTS
31
GENUINE
12465
Fraud Transactions
LIME Model Interpretation
Risk Transaction
Timeline Transaction
Fraud Transaction Every Hour
No of Transaction
Time in Hours
High
Medium