Handling Class Imbalance In Fraud Detection

A project report submitted in partial fulfillment of
the requirements for the degree of

Bachelor of Engineering
in
Computer Engineering

by

Clayton Almeida (8587)
Ron George (8605)
Akshay Naphade (8623)

Under the guidance of
Prof. Swati Ringe

DEPARTMENT OF COMPUTER ENGINEERING
Fr. Conceicao Rodrigues College of Engineering, Bandra (W), Mumbai - 400050
University of Mumbai
2021-22

This work is dedicated to my family.
I am very thankful for their motivation and support.

Internal Approval Sheet

CERTIFICATE

This is to certify that the project entitled "**Handling Class Imbalance In Fraud Detection**" is a bonafide work of **Clayton Almeida (8587), Ron George (8605), Akshay Naphade (8623)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor in Computer Engineering**.

Prof. Swati Ringe
Supervisor/Guide

Dr. Brijmohan Daga
Head of Department

Dr. Srija Unnikrishnan
Principal

Approval Sheet

Project Report Approval

This project report entitled by **Handling Class Imbalance In Fraud Detection** by **Clayton Almeida, Ron George and Akshay Naphade** is approved for the degree of Bachelor of Engineering in Computer Engineering.

Examiner 1. ————————

Examiner 2. ————————

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Clayton Almeida (8587) **(sign)** _____

Ron George (8605) **(sign)** _____

Akshay Naphade (8623) **(sign)** _____

Date:

Place:

Abstract

"Class imbalance" alludes to categorizing the false and non-deceitful classes in a dataset. The quantity of events of the positive class (minority) in class awkwardness issues is considerably more modest than the quantity of cases of the negative class (larger part) (greater part). Because of the way that the non-deceitful class is more pervasive and the false class is unprecedented, the last option would be considered exceptions, bringing about misclassification of the minority class. Subsequently, the results of inconsistent classes are not right all the time. The proposed model expects to test the dataset utilizing arbitrary undersampling procedures like Near Miss and oversampling methods like SMOTE and ADASYN, after which the fair dataset will be given to the classifier and the outcomes will be analyzed, assessed utilizing execution assessment measurements considering various essential parameters to obtain high eficiency. The results show how SMOTE technique stands out from other sampling techniques in obtaining a model with high accuracy. By resolving class imbalance problem, we can overcome various constraints in Fraud Detection, inconsistency identification, oil slick observing, spam separating, network intrusion recognition, and also in clinical applications.

Acknowledgments

We have great pleasure in presenting the report on "**Handling Class Imbalance In Fraud Detection**". We take this opportunity to express my sincere thanks towards the guide **Prof. Swati Ringe**, C.R.C.E, Bandra (W), Mumbai, for providing the technical guidelines, and the suggestions regarding the line of this work. We enjoyed discussing the work progress and working on innovative ideas on the project. We thank her for her valuable guidance and constant support in completing this project.

We thank Dr. B. S. Daga, Head of Computer Engineering Dept., Principal and the management of C.R.C.E., Mumbai for encouragement and providing necessary infrastructure for pursuing the project.

We also thank all non-teaching staff for their valuable support, to complete our project.

Clayton Almeida (8587)

Ron George (8605)

Akshay Naphade (8623)

Date:

Contents

List of Figures

List of Tables

List of Abbreviations

LR	Logistic Regression
SVM	Support Vector Machine
RF	Random Forest
NM	Near Miss Sampling
SMOTE	Synthetic Minority oversampling Technique
ADASYN	Adaptive Synthetic Sampling
RUS	Random Under Sampling

Chapter 1

Introduction

In imbalanced classes, results are not always accurate which is a very standard classification problem in machine learning. With an uneven ratio of occurrences in a class, there is always a variation in datasets. The Class Imbalance problem arises when the ratio of fraudulent activities (minority class) is very less in proportion to nonfraudulent activities (majority class). In datasets, an imbalanced class distribution occurs when one class, usually the one of more interest, the positive or minority class, is underrepresented.

- In imbalanced classes results are not always accurate which is a standard classification issue in machine learning.

- With an unequal ratio of observations in a class, there is always a disparity in datasets.

Reports of clinical diagnosis, the fintech industry, and other applications with uneven data sets are few examples. For example, in a medical diagnosis of a disease, where it is essential to identify an unfamiliar medical condition among the general population, any diagnostic mistakes can cause patients anxiety and extra difficulties. Another example is when Churn analysis is performed on real life data from a Software as a Service (SaaS) company selling an advanced cloud-based business phone system. The available dataset gathers customers data on a monthly basis and has a very imbalanced distribution of the target: a large majority of customers do not churn. The majority of real-world classification issues like spam filtering, network intrusion detection have some level of class imbalance, which means that each class does not make up an equal portion of your data set. So, it becomes very necessary to overcome such misclassifications to obtain desired results.

In datasets, a highly skewed distribution occurs when one class, usually the one of more interest, the positive or minority class, is overlooked. There are different difficulties caused by imbalance classes and they also hinder the performance of machine learning techniques. An incorrect classification can therefore lead to a major loss. As a result, it's critical for a classification model to be able to identify unusual occurrences (minority class) in datasets with a better accuracy rate. Along with accuracy, other important parameters are also to be considered for designing the appropriate model.

1.1: Motivation

Based on the review of literature for Handling Class Imbalance, we have identified the following challenges. All these constraints, the authors have come across are encouraging us to work in this domain to overcome these problems and get the desired accuracy in Handling Class Imbalance problem in Fraud Detection.

- Accuracy of a particular data [1].
- The Sampling techniques are not used appropriately which leads to results that are inaccurate

Based on the challenges, we have developed a system which will help to overcome the existing constraints and generate an accurate solution for the same. The solution considers using the sampling techniques to sample the data and compare it thereafter with each classification technique.

1.2: Objectives

1. To overcome the existing constraints that lead to inaccurate conclusions and results which might be risky in many cases.

2. To provide an accurate solution which will help the user in better decision-making process.

3. To use the classification algorithms needed to generate an accurate solution, and overcome the ambiguity in the system which will help the users to make a correct decision.

Chapter 2

Literature Review

One of the most important tasks that must be done while developing a project is literature survey. This is essentially required because of the changing needs of the world so that our creation is abreast with all the new and compatible hardware and software. This section provides a critical review of the various approaches available to handle the class imbalance problem in machine learning.

2.1: Related Work

The Authors in paper [1] have explained the issues that come with learning from imbalanced class data sets and various problems in class imbalance classification. A survey on existing approaches for handling classification with imbalanced datasets was also presented. The current trends and developments in class imbalance learning and categorization that have the ability to affect the future direction were also reviewed. They discovered that advances in machine learning approaches would mostly aid big data computing in solving many problems.

The Authors in paper [2] discovered several shortcomings of existing methods and found that the approaches designed specially to tackle the imbalance problem are not adequately effective. They assessed the effectiveness of eight machine learning approaches for detecting credit card fraud, identifying their flaws, and determining how effective they were in the case of extreme imbalance. According to the performance measures studied, the LR, C5.0 decision tree algorithm, Support Vector Machine (SVM), and Artificial Neural Network (ANN) are the top approaches (Accuracy, Sensitivity).

In paper [3] it has been explained that addressing imbalanced datasets in classification tasks is a relevant topic in research studies. The fundamental reason for this is that a typical classification algorithm may have a lower success rate when recognizing minority class occurrences. Data level strategies have been demonstrated to be the most reliable of the several solutions to this problem.

The Authors in the paper [4] have explained that the performance of fraud detection in credit card transactions is greatly affected by the sampling approaches on the dataset, selection of variables, and detection technique(s) used. In this study, a hybrid of under-sampling (the negative cases), and over-sampling (the positive cases) was carried out to achieve two setsof data distributions. The performances of the three classifiers Naïve Bayes, K Nearest Neighbor, and Logistic Regression (LR) were examined on the two sets of data distributions.

The Authors in the paper [5] have explained the undersampling which has been widely used in the class-imbalance learning area. The primary flaw in most existing undersampling approaches has been discovered to be that their data sampling procedures are heuristic-based and regardless of the classifier and evaluation metric utilized. Thus, the informative instances are discarded for the classifier during the data sampling. The key idea of this method was to parametrize the data sampler and train it to optimize the classification performance over the evaluation metrics.

In this [6], the author conducted a wide study of published papers in the last 8 years that focused on the high-class imbalance in big data in order to assess the state-of-the-art in resolving unfavorable consequences owing to class imbalance. The data- level and algorithm-level are the main methods that were discussed in this study. In order to alleviate class imbalance, data sampling methods are popular, with Random Over-Sampling approaches often producing better overall results. It also illustrates how a majority-minority classification challenge, class imbalance in the dataset(s), and prediction bias for the dominant class drastically distort the performance of classifiers.

In paper [7], a novel method that optimally adjusts the SMOTE ratios for rare classes i.e number of cases for the tuple of SMOTE ratios is too large to test all the cases is presented. For that reason, an efficient method is used in which randomly generated tuples of SMOTE ratios are used to create a model using a support vector regression (SVR). Several tuples are given as input for SMOTE ratios to the SVR model, and the best tuple of SMOTE ratios is chosen. Experimental results using the given method were significantly found to be satisfactory. The paper suggests an efficient method on how to find the SMOTE ratios that show good performance with very few tests. Hence, it dramatically reduces the amount of computations required to find the best SMOTE ratios.

2.2: Threats and Vulnerabilities in existing system

- In the published studies, there were inconsistent and conflicting results, coupled with a limited scope in evaluated techniques, indicating the need for more comprehensive, comparative studies. Also, the study showed that considering just one performance measure for imbalanced learning was misleading [3].

- The major concern is that if we increase the data it leads to increase in size and complexity [1].

- During sampling, various techniques such as oversampling, under sampling, Near Miss etc. are used where under sampling technique leads to a lot of data loss and this elimination leads to a loss of meaningful information [2].

- The sampling techniques were not used appropriately which led to inaccurate results with poor f1 score. [4]

- While classifying the fraud and the non - fraud cases, the imbalance of the classifier affects the overall performance of the system thus generating inaccurate solution [7].

Chapter 3

Problem Statement

Imbalanced class distribution in datasets generally means the number of examples from the positive class (minority) is much smaller than the number of examples of the negative class (majority). When rare instances occur seldom, they are most typically projected as rare occurrences, undetected or disregarded, or thought to be noise or outliers, resulting in more positive class (minority) misclassifications than the prevalent class. Paradoxically, the smaller class (minority) is frequently of greater significant concern, demanding a strong sense of urgency to be noticed. Most of the machine learning algorithms are biased towards the majority class & hence provide accurate results. So, there is a need to address the minority classes as well. The proposed system will perform a data-level technique called oversampling & undersampling on the dataset. After applying these techniques, balanced data will be given to the classifiers & results will be evaluated using performance evaluation metrics. Based on this, the problem statement can be proposed as follows:

To develop a system to solve the class imbalance in credit card fraud detection using resampling methods like Random Under Sampling, Near Miss techniques, ADASYN (Adaptive Synthetic), and SMOTE (Synthetic Minority oversampling Technique) to overcome the dataset's unusual events and applying machine learning algorithms to obtain more accurate findings and considering various other parameters in distinguishing the results.

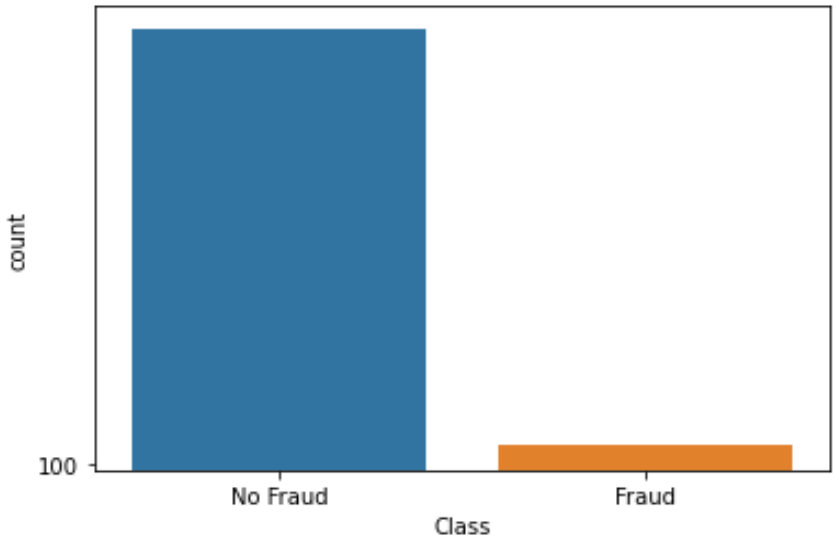

Fig 3.1 Graphical representation of Data Imbalance

Chapter 4

Project Description

4.1: Project overview

The developed system will help to overcome the existing constraints that lead to inaccurate conclusions and results which might be risky in many cases. We also understand all the limitations and the challenges that lead to poor accuracy of the system such as imbalanced data, sampling technique and classification algorithms. These challenges has motivated us to provide an accurate solution which will help the user in better decision-making process and also understand the dataset in order to generate an accurate solution.

1. **The imbalanced dataset is given to the system which is then pre-processed so that the data is in useable format for analysis.**

2. **Now we shall select any data level approach to balance the data, we will be using a sampling approach.**

3. **After sampling the data it will be given to the classifier and their performance will be evaluated.**

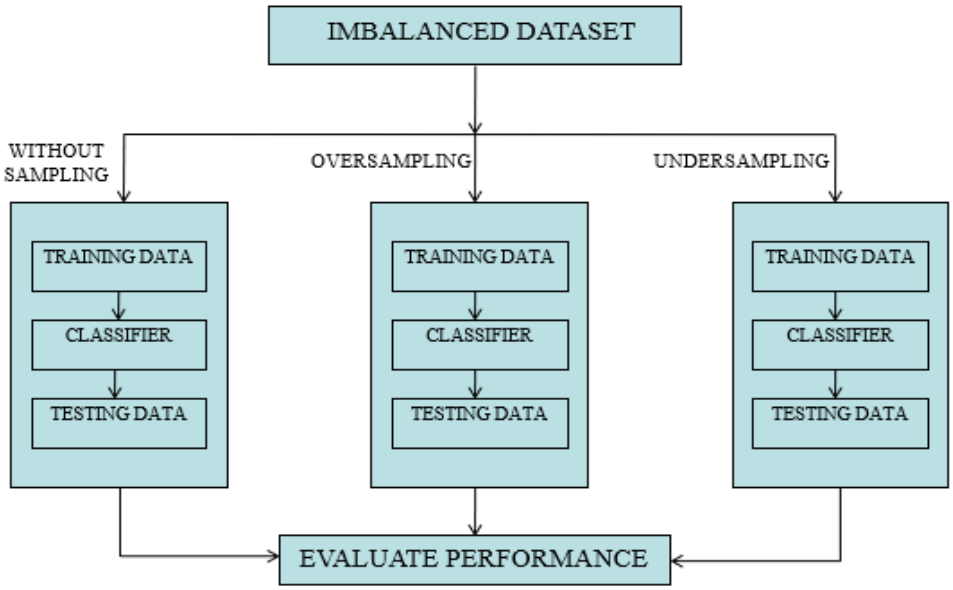

Fig 4.1 Methodology

4.2: Dataset

- Data imbalance usually reflects an unequal distribution of classes within a dataset.
- The classifier therefore tends to favor the majority class.
- The dataset contains transactions made by credit cards in September 2013 by European cardholders.
- The credit card fraud detection dataset from Kaggle consists of 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions, most of the credit card transactions are not fraud and a very few classes are fraud transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, the original features and more background information about the data cannot be provided. Features V1, V2, … V28 are the principal

components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

4.3: System Architecture

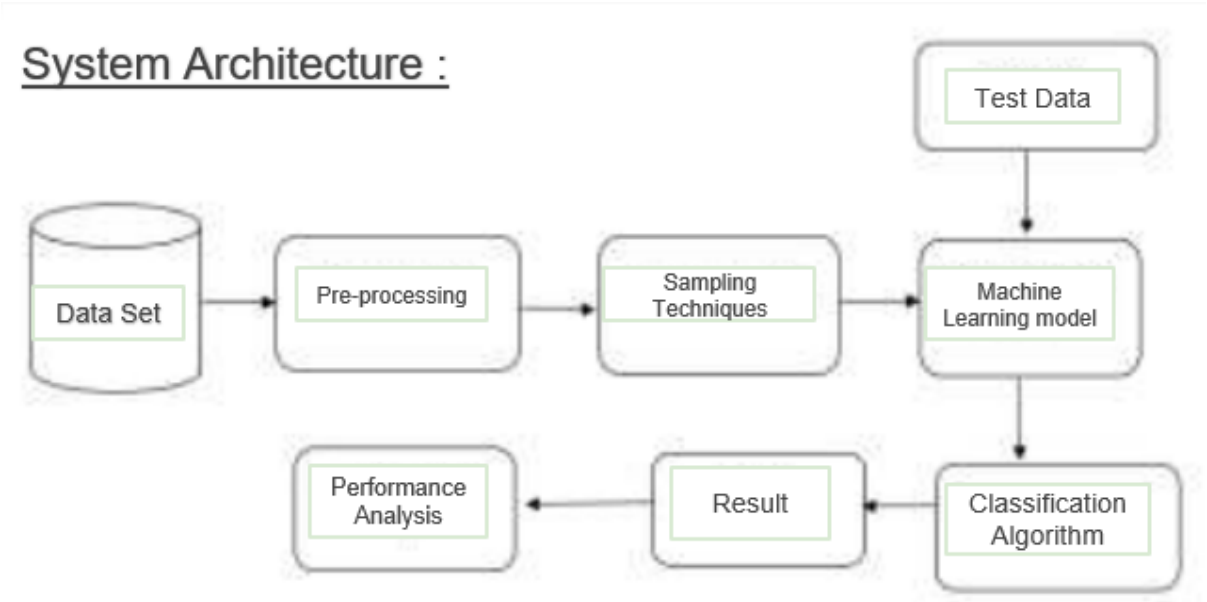

Fig 4.3 System Architecture

1.Separate the samples based on the class:

a) Dataset 1- Where all Samples belong to one class Assuming it is Majority Class.

b) Dataset 2- Where all Samples belong to Minority Class.

2.Take subsets from Majority Class without repetition.

3.Merge both the Majority sample dataset and Minority dataset for each subset drawn

4.Repeat step3 for all Datasets and build multiple models for each.

4.4: Module Description

4.4.1: Sampling Techniques

Because the dataset is highly skewed, with the number of non-frauds much outnumbering the number of frauds, it is important to resample it by lowering the majority class and raising the minority class instances. The techniques that are used in this experiment are Random oversampling which involves (Synthetic Minority Oversampling Technique) SMOTE and (Adaptive Synthetic Oversampling Technique) ADASYN while the Random undersampling technique involves the Near Miss method.

1. SMOTE

When it comes to classification challenges, the percentage of classes in the whole sample matters a lot. Imbalance is nothing but the presence of a minority class in the dataset. SMOTE is a technique for oversampling that tries to generate fresh synthetic observations or data points. It entails locating the feature vector and its closest neighbors, as well as determining the difference between them. The difference is multiplied by a random value between 0 and 1 before plotting the new points.

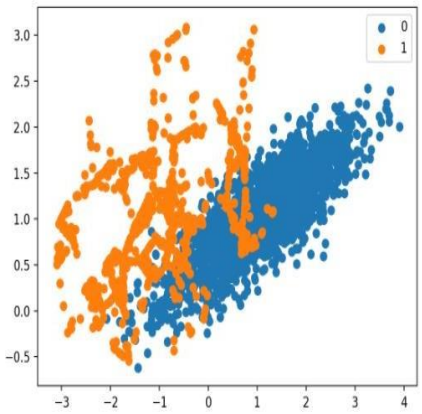

Fig 4.4 Smote technique

2. Near Miss

The term "near miss" refers to a sampling approach that reduces the number of samples taken from the majority class. Using a distance instead of sampling the Minority class will make the majority class equal to the minority class.

The majority class samples are chosen to be very close to the minority class observations in the Near Miss -1 approach, and the majority class is undersampled.

The second technique, NearMiss-2, picks majority class samples with the least average distances to the three most distant minority class samples. For each minority class sample, the third method NearMiss3 removes a set number of the nearest majority class samples.

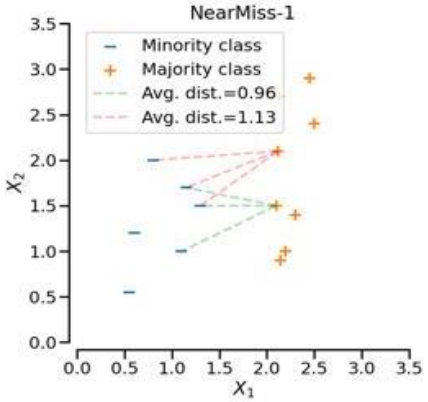

Fig 4.5 Near Miss technique

3. ADASYN

ADASYN is a method for generating synthetic data samples for minority class observations, thereby reducing bias and misclassifications that might impair data correctness. By altering the classifier decision boundary, ADASYN aids in improving learning performance. It uses the density distribution approach to determine the number of samples required in the minority class in order to match the number of observations in the majority class.

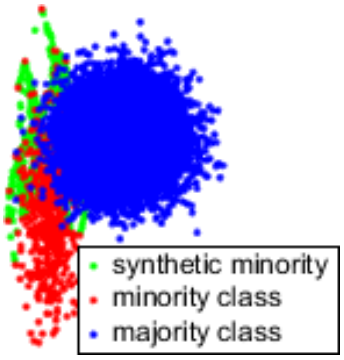

Fig 4.6 ADASYN technique

4.4.2: Classification Algorithms

After the data is balanced, classification algorithms are used to classify the data into fraud or non-fraud. We have used classification algorithms such as Logistic Regression (LR) and Random Forest.

Logistic Regression – It is used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, where there would be only two possible classes. Logistic regression tries to find the optimal decision boundary that best separates the classes. An advantage of logistic regression is that it is easy to implement and yet provides great training efficiency. Training a model with this algorithm does not require high computation power. This model (algorithm) will predict whether a new transaction is fraudulent or not. The Credit Card Fraud Detection problem is of significant importance to the banking industry because banks each year spend hundreds of millions of dollars due to fraud. When a credit card transaction happens, the bank makes a note of several factors. For instance, the date of the transaction, amount, place, type of purchase, etc. Based on these factors, they develop a Logistic Regression model of whether or not the transaction is a fraud. For instance, if the amount is too high and the bank knows that the concerned person never makes purchases that high, they may label it as a fraud.

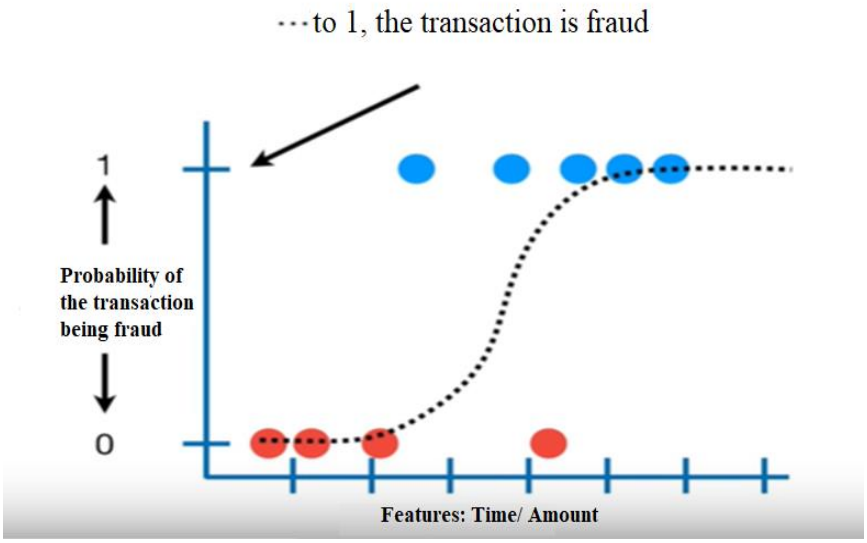

Fig 4.7 Logistic Regression

The above Fig 3.2. is an example of logistic regression where an S shaped curve is plotted ranging between 0 and 1 to predict the probability of the credit card transaction being fraud or non-fraud based on the parameters such as Time and Amount.

Random Forest - Random forest, one of ensemble methods, is a combination of multiple tree predictors such that each tree depends on a random independent dataset and all trees in the forest are of the same distribution [20]. The capacity of random forest not only depends on the strength of individual tree but also the correlation between different trees. The stronger the strength of single tree and the less the correlation of different tress, the better the performance of random forest [15]. The popularity of decision tree models [23] in data mining is owed to their simplification in algorithm and flexibility in handling different data attribute types.

Steps for performing Random Forest:

- Step 1: Create a Bootstrapped Data Set. Bootstrapping is an estimation method used to make predictions on a data set by re-sampling it.

- Step 2: Creating Decision Trees.

- Step 3: Go back to Step 1 and Repeat.

- Step 4: Predicting the outcome of a new data point.

- Step 5: Evaluate the Model.

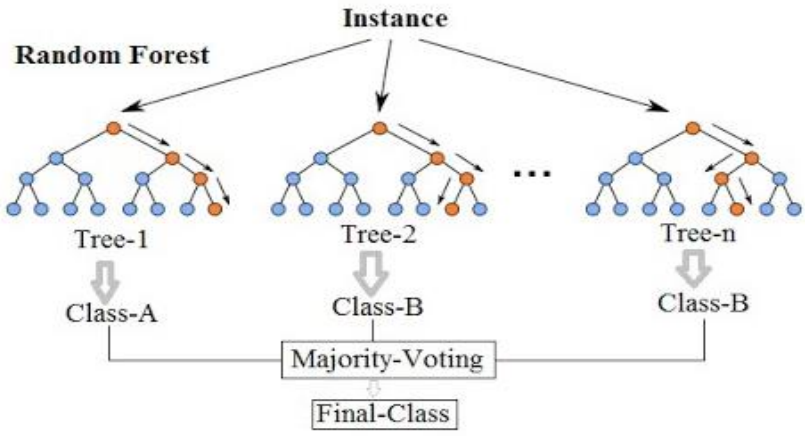

Fig 4.8 Random Forest

For e.g., Random forest is like bootstrapping algorithm with Decision tree (CART) model. Random forest tries to build multiple decision trees with different samples and different initial variables. For instance, it will take a random sample of 100 observation and 5 randomly chosen initial variables to build a CART model. It will repeat the process (say) 10 times and then make a final prediction on each observation. The data will be run down each decision tree and whichever class gets the maximum number of votes Class A or Class B, will be the final prediction.

4.4.3: Performance Evaluation Metrics

To evaluate the performance of proposed system, various standard classifier evaluation metrics such as precision, recall, are used. These are explained as follows:

a) Accuracy can be calculated as the ratio of number of correctly predicted reviews to the number of total number of reviews present in the corpus.

b) Precision is the exactness of the classifier.

$$\frac{True\ Positive\ (TP)}{True\ Positive(TP)+False\ Positive(FP)}$$

c) Recall is the process of measuring the completeness of classifier

$$\frac{True\ Positive(TP)}{True\ Positive(TP)\ +\ False\ Negative(FN)}$$

Chapter 5

System Design and Requirements

5.1: Jupyter Notebook

As a server-client application, the Jupyter Notebook App allows you to edit and run your notebooks via a web browser. The application can be executed on a PC without Internet access, or it can be installed on a remote server, where you can access it through the Internet. Its two main components are the kernels and a dashboard.

A kernel is a program that runs and introspects the user's code. The Jupyter Notebook App has a kernel for Python code. The dashboard of the application not only shows you the notebook documents that you have made and can reopen but can also be used to manage the kernels: you can which ones are running and shut them down if necessary

5.2: Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

5.3: VS Code

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

5.4: Hardware Requirements

The hardware requirements for the project are:

- Processor required is Intel Core i3 or Higher core Processor (CPU)/2.1GHz or Higher.
- RAM Minimum of 512 MB, 4 GB is recommended.
- Storage Between 850 MB and 1.2 GB, depending on the language version.
- Minimum screen resolution required is 240 x 320.

5.5: Software Requirements

The software requirements for the project are:

Python : The code is written in Python either using Anaconda IDE or jupyter notebook and the code has been applied to a number of test cases for analysis. Python 3.8.3 is used for project development and analysis.

Algorithms : We have used Machine Learning Algorithms like Logistic Regression, Random Forest, SVM, etc to evaluate the performances of the model.

Pandas 1.3.4: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Imblearn 0.7.0 : Imbalanced-Learn is a Python module that helps in balancing the datasets which are highly skewed or biased towards some classes. Thus, it helps in resampling the classes which are otherwise oversampled or undersampled.

Seaborn 0.11.2 : Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

NumPy 1.20.3 : NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Yellowbrick 1.4 : Yellowbrick is a machine learning visualization library. Yellowbrick is mainly designed to visualize and diagnose the machine learning models. It helps in the model selection process, hyperparameter tuning and algorithm selection.

Matplotlib 3.4.3 : Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Chapter 6

Implementation Details

6.1: Methodology

In this study we are going to use various sampling techniques like oversampling and under sampling thereby balancing the data and then giving it to the classifier after which the results will be compared to check which resampling techniques gives accurate results.

6.2: Implementation

Plotting the data to check Class Imbalance:

1. First the dataset creditcard.csv is loaded

2. The graph is being plotted showing the number of frauds and non-frauds

3. Figure shows the data imbalance where the number of frauds (Class 1) are very less when compared to non-frauds (Class 0)

Code:

```
sns.countplot('Class', data=df, palette=colors)
plt.title('Class Distributions \n (0: No Fraud || 1: Fraud)', fontsize=14)
plt.show()
```

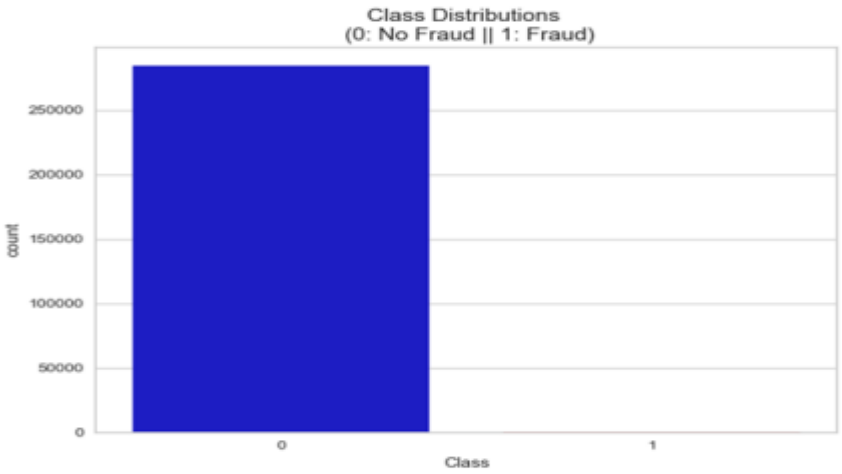

Fig 6.1 Distribution of Fraud and Non-fraud cases

Output:

No Frauds: 99.83 % of the dataset

Frauds: 0.17% of the dataset

Removing outliers

```
v12_fraud = new_df['V12'].loc[new_df['Class'] == 1].values
q25, q75 = np.percentile(v12_fraud, 25), np.percentile(v12_fraud, 75)
v12_iqr = q75 - q25
```

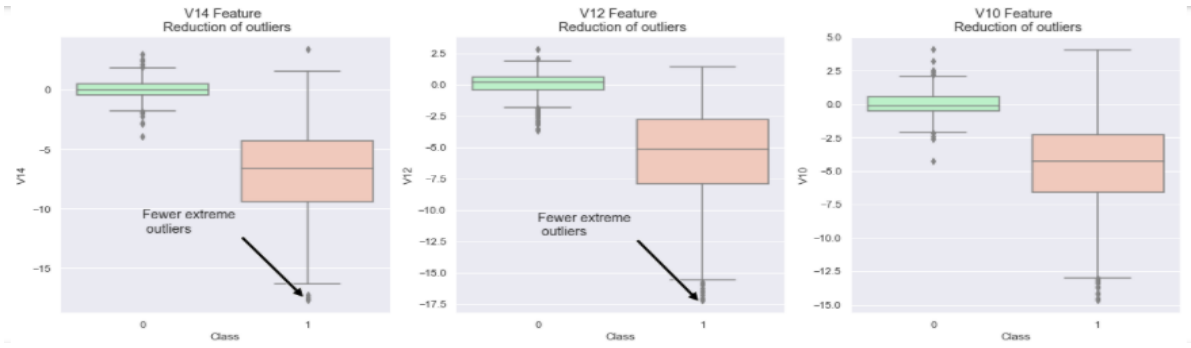

Fig 6.2 Reduction of outliers

Without Sampling

Classification algorithms like SVM, Random Forest, K Neighbours Classification and Logistic Regression were applied on the training dataset before applying the sampling technique and the accuracies were calculated for each one of the classification techniques.

The accuracies for each one of them are

Logistic Regression has a 94.0 % accuracy score.
KNN Classifier has a 93.0 % accuracy score.
SVM has a 93.0 % accuracy score.
Random Forest Classifier has a 93.0 % accuracy score.

Sampling the Dataset

The dataset being highly imbalanced, the results are biased towards the majority class. Hence it is necessary to sample the dataset to get better results in terms of accuracy, precision and recall. The dataset is sampled using Random Under sampling and Oversampling Techniques to balance the majority and the minority class.

1. Near Miss Sampling Technique

Near Miss Sampling Technique aims to balance class distribution by randomly eliminating majority class examples. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes.

Code:

```
X, y = NearMiss().fit_sample(undersample_X.values, undersample_y.values)
dataframe=pd.DataFrame(y, columns=['target'])
target_count = dataframe.target.value_counts()
print('Class 0:', target_count[0])
print('Class 1:', target_count[1])
```

```
zero=target_count[0]
one=target_count[1]
left = [1, 2]
height = [zero,one]
tick_label = ['Not Fraud', 'Fraud']
plt.bar(left, height, tick_label = tick_label, width = 0.8, color = ['red', 'green'])
plt.xlabel('x - axis')
plt.ylabel('y - axis')
plt.title('Nearmiss')
plt.show()
lg= LogisticRegression().fit(X, y)
Y_Test_Pred = lg.predict(original_Xtest)
```

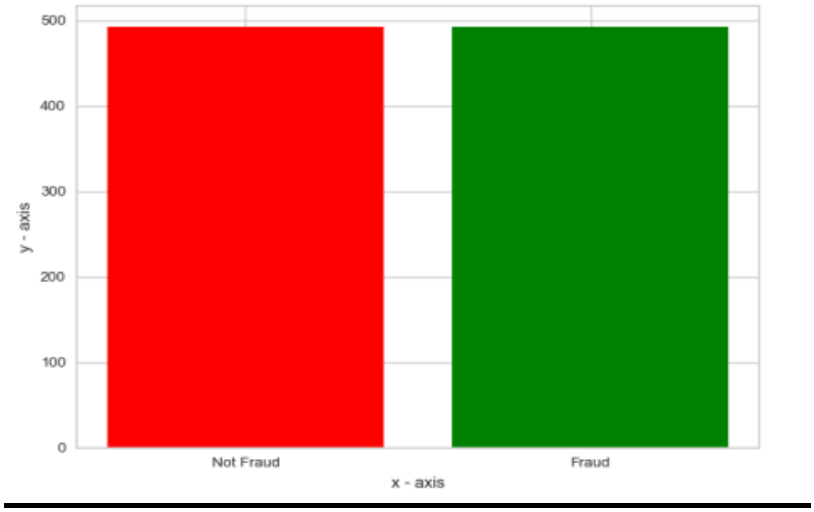

Fig 6.3 Data Distribution after Near Miss Sampling

Classification Algorithm

Logistic Regression Classification Technique is used after sampling the dataset and the confusion matrix is shown

Code:

```
# Confusion Matrix
matrix =confusion_matrix(original_ytest, Y_Test_Pred)
class_names=[0,1]
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
sns.heatmap(pd.DataFrame(matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1)
#plt.title('nearmiss Svm', fontsize=8)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.show()
```

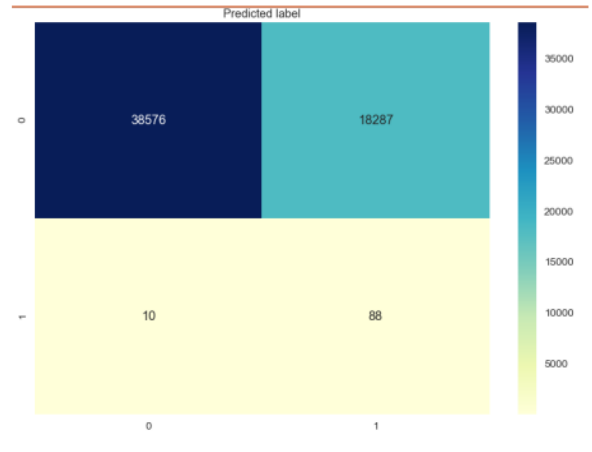

Fig 6.4 Confusion Matrix

Chapter 7

Results and Conclusion

7.1 Result

In this dataset we have 492 out of 2,84,807 which are fraud transactions. That's only 0.173% of all of the transactions in this dataset. Data is not balanced because less amount of fraud cases as compared to huge transaction data. For all datasets, 70% of the data is kept for the training and validation while 30% is used for the testing purpose.

Classification algorithms like SVM, Random Forest, KNN Classification, and Logistic Regression were applied to the training dataset before applying the sampling technique and the accuracies were calculated for each one of the classification techniques. The accuracies for each one of them are as follows: Logistic Regression has a 94.0 % accuracy score. KNN Classifier has a 93.0 % accuracy score. SVM has a 93.0 % accuracy score. Random Forest Classifier has a 93.0 % accuracy score.

However, the results are unsatisfactory because the majority of class observations are non-fraud and just 5% are a fraud; hence, these findings cannot be described as trustworthy. As a result, balancing the dataset is critical in order to reduce errors or misclassifications.

We have balanced the dataset using the resampling techniques and then used each one of them with the ML Algorithms and the results were compared.

a. Comparing the performances of Near Miss Sampling Technique along with Logistic Regression and Random Forest.

Table 7.1 Near Miss LR

Predicted	Actual	
	0	1
0	38576	18287
1	10	88

Table 7.2 Near Miss RF

Predicted	Actual	
	0	1
0	13922	42948
1	0	98

Observation – **In case of near miss sampling using LR and RF, the accuracy using both the algorithms was quite low and the confusion matrix are shown above.**

b. Comparing the performances of Random Under sampling with Logistic Regression and Random Forest.

Table 7.3 Random Under sampling LR

Predicted	Actual	
	0	1
0	54661	2202
1	8	90

Table 7.4 Random Under sampling RF

Predicted	Actual	
	0	1
0	55661	1202
1	0	98

Observation – **In case of random under sampling using LR the accuracy was a respectable 97.61% and while using RF, the accuracy was 97.94% which is slightly better and the confusion matrix is shown above.**

c. Comparing the performances of SMOTE with Logistic Regression and Random Forest.

Table 7.5 SMOTE LR

Predicted	Actual	
	0	1
0	56232	631
1	13	85

Table 7.6 SMOTE RF

Predicted	Actual	
	0	1
0	55661	1202
1	33	65

Observation – **The results of SMOTE with LR and SMOTE with RF are shown below and SMOTE with RF has performed well with an accuracy of 99.94% and the confusion matrices are shown in the above tables.**

d. Comparing the performances of ADASYN with Logistic Regression and Random Forest.

Table 7.7 ADASYN LR

Predicted	Actual	
	0	1
0	54817	2046
1	1	88

Table 7.8 ADASYN RF

Predicted	Actual	
	0	1
0	56862	1
1	34	64

Observation – **From the above table we observed the confusion matrices of ADASYN sampling technique using LR and RF algorithms and the accuracy was 96.38% and 99.93% respectively.**

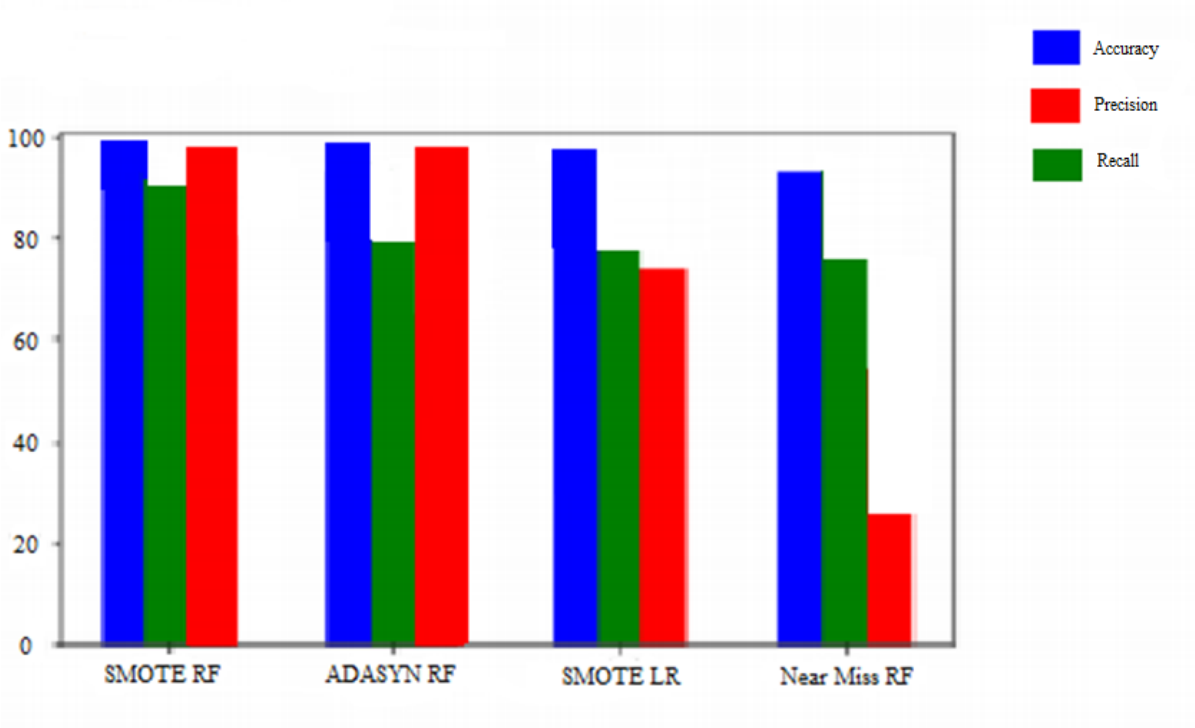

Fig 7.1 Comparitive Bar Graph of results

The above Figure shows a comparison between SMOTE RF, ADASYN RF, SMOTE LR and Near Miss RF in terms of Accuracy, Precision and Recall. SMOTE sampling technique used along with Random Forest and ADASYN technique along with Random Forest performed the best in terms of accuracy with a score of 99.93% and 99.94% respectively when compared to SMOTE LR and Near Miss RF with an accuracy score of 98.86% and 24.87% respectively. In terms of precision Near Miss RF performed the best while SMOTE RF performed best in terms of recall.

Comparative Analysis of accuracy, precision, recall after applying sampling techniques on the dataset.

Table 7.9 Comparitive Results of various techniques

Method	Accuracy	Precision	Recall	AUPRC
Near Miss LR	67.88 %	0.89	0.47	0.02
Near Miss RF	24.87 %	0.98	0.03	0.72
Random Under sampling LR	97.61 %	0.89	0.61	0.79
Random Under Sampling RF	97.94 %	0.99	0.09	0.89
SMOTE LR	98.86 %	0.87	0.12	0.78
SMOTE RF	99.94%	0.68	0.98	0.81
ADASYN LR	96.38 %	0.89	0.04	0.79
ADASYN RF	99.92 %	0.64	0.98	0.79

We now know everything there is to know about all of the techniques and classifiers, and we've done a comparison of all of the machine learning algorithms. We completed all of the processes outlined in the block diagram and then compared the results of each machine learning algorithm when applied to various sampling approaches. To overcome the problem of class imbalance, we applied the data-level approaches of oversampling and the undersampling and compared the results of each. We employed the Area Under Curve (AUC) for model evaluation, which allowed us to determine whether our model performed better after data balancing. The results demonstrate that ADAYSN and RF, as well as SMOTE and RF, produced the greatest results, with 99.93 % accuracy. The accuracy before using the sampling technique was 96% but our aim was to get a higher accuracy as the number of frauds is far less when compared to non-fraud cases. As a result, while the accuracy was greater, the precision and recall were significantly worse when compared to the after-sampling technique. They did, however, have certain flaws in terms of precision and memory.

7.2 Conclusion

Most firms, notably those in banking (primarily credit card fraud detection), fintech, retail, and e-commerce, have had significant hurdles in detecting scams. Any fraud has a detrimental impact on an organization's bottom line, reputation, and the willingness of prospective prospects and existing customers to transact with it. It will also be beneficial to the organization so that they can prevent such types of frauds in the future and any misleading results.

By handling the class imbalance problem, we can also overcome various other constraints in the medical sectors and also in commercial sectors. We have successfully compared the performances of the classification algorithms with each sampling technique to identify the shortcomings with every pair of classification algorithms with the sampling technique.

The purpose of this research was to test if the classification model could discriminate between fraudulent and non- fraudulent transactions, as well as to determine which sampling procedures might aid the model's performance. The classification model's performance was better than it had been before employing the sampling strategy. Our tests revealed that combining the ADASYN method with RF and SMOTE with RF yielded 99.94% encouraging results. They will be combined with other machine learning techniques in the future to provide better accuracy.

Despite the fact that these strategies improve the performance of the classifier, many fraud cases continue to be undetected. In addition, our research revealed that using only one performance measure to assess uneven learning is deceptive. In the future, different optimization techniques can be identified and experimented with. Also, the effects of other sampling approaches can be investigated along with other classification algorithms.

7.3 Applications

Class Imbalance appear in many domains, including:

Fraud detection : If we are trying to identify the fraudulent transactions from a dataset, we won't have a sufficient number. Hence, the model may try to fit the majority class leading to biased prediction and at the same time also provide misleading accuracy.

Spam filtering : Spam filtering is one such application where class imbalance is apparent. There are many more non-spam emails in a typical inbox than spam emails.

Disease screening : In a medical diagnosis of a rare disease where there is critical need to identify such a rare medical condition a classification model should be able to achieve higher identification rate on the rare occurrences (minority class) in datasets.

SaaS subscription churn : Churn analysis is performed on real life data from a Software as a Service (SaaS) company selling an advanced cloud-based business phone system, Aircall. The available dataset gathers customers data on a monthly basis and has a very imbalanced distribution of the target: a large majority of customers do not churn.

Network intrusion detection : In intrusion detection systems, attack patterns or malicious activities can be classified by monitoring the network where the number of instances of attacks is comparatively much smaller than the regular network traffic.

7.4 Future Scope

Fraud detection has been one of the major challenges for most organizations particularly those in banking mainly credit card fraud detection, finance, retail, and e-commerce. This goes without saying that any fraud negatively affects an organization's bottom line, its reputation and deter future prospects and current customers alike to transact with it. It will also be beneficial to the organization so that they can prevent such type of frauds in future and

any misleading results. By handling class imbalance problem, we can also overcome various other constraints in medical sectors and also in commercial sectors.

In future different optimization techniques can be identified and experimented. Also, effects of other sampling approaches can be investigated along with other classification algorithms. In future work, we intend to enhance the performance and take the security and privacy of the data in real time into consideration

References

[1] Awoyemi J. O, Adetunmbi A. O & Oluwadare S. A, "Credit card fraud detection using machine learning techniques: A comparative analysis," International Conference on Computing Networking and Informatics (ICCNI), doi:10.1109/iccni.2017.8123782, 2017

[2] Aida Ali1, Siti Mariyam Shamsuddin, and Anca L.Ralescu, "Classification with class imbalance problem: a review,"Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, November 2015.

[3] Makki S, Assaghir Z, Taher Y, Haque R, Hacid M.S & Zeineddine H, "An Experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection," IEEE Access, 1–1, doi:10.1109/access.2019.2927266, 2019

[4] Hartono, Hartono & Sitompul, Opim & Tulus, Tulus & Nababan, Erna. (2018). "Biased support vector machine and weighted-SMOTE in handling class imbalance problem." International Journal of Advances in Intelligent Informatics. 4. 21. 10.26555/ijain. v4i1.146. Springer, Singapore,2018

[5] Zhu B, Baesens B, Backiel A & Vanden Broucke S.K, "Benchmarking sampling techniques for imbalance learning in churn prediction", Journal of the Operational Research Society, 69(1), 49-65, 2018.

[6] Leevy J. L, Khoshgoftaar T. M, Bauder R. A & Seliya N, "A survey on addressing high-class imbalance in big data," Journal of Big Data, 5(1), 42, 2018.

[7] Machine-Learning Approach to Optimize SMOTE Ratio in Class Imbalance Dataset by Jae- Hyun Seo 1 and Yong-Hyuk Kim 2 1 Department of Computer Science and Engineering, Wonkwang University, 460 Iksandae-ro, Iksan-si, Jeonbuk 54649, Republic of Korea.

[8] Wei Wei, Jinjiu Li, Longbing Cao, Yuming Ou & Jiahang Chen, "Effective detection of sophisticated online banking fraud on extremely imbalanced data," World Wide Web, 16(4), 449-475,2013

[9] Zhang, J., Mani, I.: "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction." Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, (2003).

[10] Khor K.C, Ting C.Y & Phon-Amnuaisuk S, "The effectiveness of sampling methods for the imbalanced network intrusion detection data set," In Recent Advances on Soft Computing and Data Mining (pp. 613-622), Springer,2014

[11] Difficulty Factors and preprocessing in Imbalanced Datasets: An Experimental study on Artificial Data Szymon Wojciechowski, Szymon wilk.

[12] Credit Card Fraud Detection Dataset: https://www.kaggle.com/dalpozz/creditcardfraud Towards data science. He, Haibo & Bai, Yang & Garcia, Edwardo & Li, Shutao. (2008). "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning." Proceedings of the International Joint Conference on Neural Networks. 1322 - 1328. 10.1109/IJCNN.2008.4633969.

[13] Improving electric fraud detection using class imbalance Strategies Mat´ıas Di Martino, Federico Decia, Juan Molinelli and Alicia Fern´andez Instituto de Ingenier´ıa El´ectrica, Facultad de Ingenier´ıa Universidad de la Rep´ublica Montevideo, Montevideo, Uruguay.

[14] C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure Nitesh V. Chawla, Customer Behavior Analytics, Business Analytic Solutions, CIBC, BCE Place, 11th Floor, 161 Bay Street, Toronto, ON, CANADA M6S 5A6

[15] An Empirical Study of AML Approach for Credit Card Fraud Detection–Financial Transactions A. Singh, A. Jain, University School of Information, Communication & Technology Guru Gobind Singh Indraprastha University, Delhi, India.