A predictive model to minimize false-positive declines in Electronic Card Not Present financial transactions using feature engineering techniques

A Thesis Submitted for the Degree of Master of Computer Science

D. M. S Delgolla

University of Colombo School of Computing

2020

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information that have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: D.M.S Delgolla

Registration Number:2018/MCS/-13

Index Number:18440131

_____

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. /Ms. _____ under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by,

Supervisor Name: Dr. Thilina Halloluwa

_____

Signature of the Supervisor & Date

ACKNOWLEDGEMENTS

This project would not have become possible without the support and guidance of many people.

First of all, I would like to convey my sincere gratitude to the supervisor of the project, Dr Thilina Halloluwa, Lecturer of University of Colombo School of Computing(UCSC), Mr. Upul Rathnayake, the core-supervisor and all the UCS staff for the continuous support and the guidance given out throughout the research.

I also like to convey my special thank to Dr. Lasanthi Silva, who was the coordinator of the Master's research projects for all the support given out throughout the research program.

I would also like to extend my sincere gratitude to Mr. Knaniska Weeramunda and Mr. Sasindu Pathiranage Co-founders of DirectPay for providing the data sets and encouraging me to research the fintech domain.

I also like to make this an opportunity to thank all my master's colleagues and friends who extend their thoughts and experience on research works and related studies.

It's my great pleasure to acknowledge all the evaluators and examiners for sharing their honest reviews and valuable insights from the very beginning of the project.

ABSTRACT

In this research, we have proposed a predictive model to minimize false positive ("Legitimate transactions are being declined falsely identifying as fraudulent") declines in electronic CNP transactions. Related to the increased popularity of digital payments FP declines are becoming a severe problem among merchants who provide digital payment solutions. It's estimated that nearly 10% of the transactions are been declined as fraudulent transactions but only very few of them have fallen into the fraud category. To address this problem we have proposed a feature engineering technique based on behavior analysis. Our research is conducted based on a real-life CNP transactional data set from one of the largest fintech service solution providers in Sri Lanka and we have generated 130 features for each transaction and have employed an XG Boost to learn the classifier and found out that performances of the xgBoost classifier has shown nearly 6% improvement in the F-Score and obtained 0.996 for the AUC after the application of feature engineering techniques. We found out that this solution can mainly benefit the merchants who provide electronic payment solutions which involve CNP transactions to minimize false-positive declines targeting legitimate frequent customers and by the same, it minimizes the fraud losses and protects the customer's interests.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

APATE –Anomaly Prevention using Advanced Transaction Exploration

CNP – Card Not Present

CVV – Card Verification Value

FDC- Federal Trade Commission

FN – False Negatives

FP – False Positives

HMM- Hidden Marcow Model

HOBA – Homogeneity Oriented Behavior Analysis

IPG – Internet Payment Gateway

MPGS – Mastercard Payment Gateway System

mPos – Mobile Point Of Sale

RF - Random Forest

RFM – Recurrent Frequent and Monetary framework

SAAS – Software As A Service

SVM – Support Vector Machine

CHAPTER 1

INTRODUCTION

Financial technology is the emergence of innovation and technology in financial services or companies to serve the financial services which are provided by the business to consumers. With the emergence of financial technologies (fintech), society is transforming more into digital payment solutions, which include fund transfers, online payments, and digital banking. And it has shown significant growth in fintech applications over the last few years. Digital payments are based on the card, not present transactions so that it has become a prominent target of cybercriminals as it's more difficult for merchants to figure out who the transaction is making

According to a survey conducted by the Federal Trade Commission (FDC)(*Consumer Sentinel Network,* n.d.). It has shown that Credit card fraud has seen unprecedented growth in recent months and has become the fastest-growing form of identity theft. Moreover, the same report shows that out of 1.7 million fraud reports, 23% indicated that money was lost. Furthermore, in 2019, people reported losing more than $1.9 billion to fraud, which shows a significant increase of over $293 million what was reported in 2018. This is the accumulated loss due to actual fraud and the loss that is caused due to false fraud detection costs as much as the approved amount, which is usually a hundred times larger, furthermore, the loss due to missed fraud may continue to occur until it is detected(Kim et al., 2019).

By 2021 online buyers have increased by 62% ("A Major Challenge - False Positives," 2020) and the impact of COVID-19 has further accelerated the transformation to digital payments and at the same time e-commerce industry and CNP transactions have also experienced a sharp rise in cyberattacks("Vesta 2021"). With regards to the acceleration of electronic payment methods, the number of frauds and fraudulent attempts is reported to increase dangerously high. To overcome this problem banks and the card authorities have applied multilayered fraud detection systems to strengthen their security but this has ultimately created a new challenge causing to increase the False Positives, which simply means "A legitimate customer sale is declined flagged as a fraud" According to (Grandi, 2021) it reveals that this e-commerce industry will experience a loss of $443 billion by 2021 due false positive declines, This loss is very much larger than the projected fraud loss of $6.4 billion.

When considering the impact of false positives it has long-term effects where the false positives go unnoticed as an online business perceives them as successfully thwarted fraud attempts instead of forgone sales. False Positives harm online business financially in four main fundamental ways ("Fraud.com. 2021.")

1. Immediate Revenue Lost – Any order or payment that has been wrongly turned down is revenue that's not realized.

2. Lost customer lifetime value – Lifetime customer value is the total profit anticipated from all future purchases by a customer. Legitimate customers who are wrongly rejected will often stop buying from that particular merchant permanently.

3. Wasted acquisition spends – This refers to all the costs associated with convincing a customer to place an order E.g. research, marketing, and advertising expenses. If the merchant has spent $10 to convince the customer to buy but mistakenly declined their order (false positive) then that $10 acquisition cost has fallen into the lost revenue.

4. Degraded brand image – False positives can cause intangible loss where the customers can share their bad experience with social media, viral posts, and bad customer reviews can cause them to reach thousands of potential customs. The impact caused by intangible losses is difficult to quantify and the negative publicity creates a bad image of the provider.

A credit card fraud can cause tangible and intangible losses to a business. A loss of money is a tangible loss while a bad customer experience is an intangible loss which can cause the same harm as a tangible loss as they communicate their bad experience with peer fellows so according to this increase of the declining online card transactions for false fraud alms is also a problem as it's not being a fraud and that allows to decrease the customer satisfaction and the customers tend to move for alternative options making it an intangible loss. With this, it can be seen that False Positives are becoming a bigger problem as same as the actual frauds in online payment systems, so it's important to distinguish the actual fraud attempts from the false alarms in business decision making and strategic development.

1.1 Motivation

In card-present transactions, as the name implies the transaction occurs with the presence of the card and the person. The customer may present at the location of the transaction and it incurs cash withdrawals, payment for supermarkets, etc., and those transactions are usually made through a terminal, a magnetic stripe card reader, a chip reader, or through a contactless reader. The behavior of these transactions is completely different from a card, not a present transaction. Card present transactions are less risky compared to the card, not present transactions because they need to have a physical card, and location and terminal details are usually recorded and put into account.

In card, not present transactions it has to enter only the card details to proceed with the payment, and Electronic payments fall into this category. For these types of transactions, there's no need to have a physical card and card, not present transactions involve transactions made via the phone, online payments, recurring and subscription payments, online invoices, etc.

The data set that we use in this research is unique. We use an electronic payment transaction data set collected from our industrial partner DirectPay, one of the leading fintech payment solution providers in Sri Lanka. Our data set contains the transactions made using mobile wallets, Mobile point of sale (mPos), and Internet Payment Gateway (IPG) where all the transactions are fallen into the card, not present transaction category. And we are using an application-level transactional data set. This data set has less dimensionality compared to the banker data set but it's rich with non-card present electronic payment transactional data set.

When banks experience the growing fraud risk to mitigate the fraud risk they apply some strategies and techniques which sometimes result in increased false positives. False-positive declines significantly and badly affect the application-level service providers. Digital retailers mainly give more attention and weight to false positives than they do to actual frauds because this transaction data set consists of one-time filtered data set through the banker's fraud detection systems. When considering the DirectPay transactional data set we have experienced that nearly 10% of the total transactions have been failed due to the "Do not honor" response. Banks decline transactions providing "Do not honor" when they suspect it as a fraudulent transaction and they do not reveal the actual reason for rejection. Currently, the 24/7 monitoring center examines the transactions and does a manual inspection when a customer arouses a complaint. This process involves high costs due to the high level of human involvement. Even though there exist such manual inspection procedures some transactions go unnoticed resulting in the loss of valuable customers.

To overcome those above-mentioned problems we propose an improved machine learning solution followed by automated feature engineering techniques to overcome the issues caused due to false positives and “Do not honor” responses which are now becoming a severe problem among electronic payment solution providers. This kind of solution incurs not only financial benefits but also improves the effectiveness of the 24/7 alert management systems. Through this research, the fintech organizations that provide online payment solutions and customers who engage in online transactions are encouraged to engage more with electronic payments enabled systems and embrace new technologies. Ultimately this approach will provide a secure environment and minimize the loss that is caused due to tangible and intangible means of damages.

The expected contribution to the field of computing by conducting this research are summarized below

- Address the issue of false-positive declines in electronic payment systems which is now becoming a severe problem.
- Introduce a novel feature engineering technique followed by machine learning techniques to minimize the possible fraudulent attempts in online transactions.

1.2 Research problem

1.2.1 Main research question

How to minimize the false positives in electronic payment systems which involve CNP transactions?

As I have mentioned in the Motivation section from the merchants' perspective there’s a 62% rise of false-positive declines compared to the previous years and some merchants are still unaware of this problem. DirectPay can be identified as one of these e-commerce payment solution providers in Sri Lanka. The DirectPay data set contains only card, not present electronic payment transactional data set, and by the same, due to the increasing popularity of digital payments the main focus was on fraud detection and reducing the false negatives, and as a result, the rise of false positives have gone unnoticed. As a result, the false-positive rate also has been increased, which is becoming a severe problem for electronic payment systems. DirectPay data set consists of a transactional data set that’s one time filtered out by the bank’s fraud detection system and we have experienced that nearly 10% of the total transactions have been

declined due to “Do not honor”. In most cases “Do not honor” response is issued by the bank when they reject the transaction for suspicious behavior of fraud. Receiving “Do not honor” and declining legitimate transactions is becoming a severe problem that leads to the disappointment of loyal customers. So In our research, we mainly focus on identifying false positive declines in the electronic payment systems

1.2.2 Sub research questions

1. How to identify the most effective features to summarize user behaviors?
   It’s essential to identify the most effective and efficient features that are required to summarize the user’s behaviors out of the hundreds of features contained in the data set.
2. How to identify the best customers related to e-commerce Card Not Present (CNP) payment platforms?

   According to (“Fraud.com. 2021.”) It has said that 40m% of European consumers have said that they won’t do business again with a merchant which has declined their card once it’s been a legitimate transaction therefore the increase of false positives increases the negative feelings of the customers towards the merchants so that it can greatly affect the firm’s best clients. Currently, there's no proper mechanism to identify the best customers in an electronic payment system related to their transactional behaviors. According to (“CNP White Papers. 2018”) 70% of the merchants are much concerned about the false positives and they have been told that there’s less likely to occur a fraudulent transaction from a frequent customer. Through these facts, we can see that it’s necessary to identify the best clients through their spending habits.

1.3 Research Aims and Objectives.

1.3.1 Aim

Our research aims to propose a novel feature engineering technique to minimize plausible false fraud declines in electronic payment systems.

1.3.2 Objectives

1. Minimize the plausible losses caused for e-commerce service providers due to false-positive decline in electronic payment systems.
2. Minimize the plausible fraudulent attempts in electronic payment systems analyzing user behavior.

1.4 Scope

The research would be conducted using electronic payment transactions data set obtained from DirectPay. The ultimate goal is to propose a novel feature engineering technique followed by machine learning techniques to minimize false-positive declines in electronic payment systems

The following results would be achieved at the end of the research

1. A set of features derived from basic features.
2. A set of rules used for feature engineering
3. Evaluate the performance of the features in the presence of different machine learning models
4. A predictive model to minimize the false-positive declines in electronic payment systems

1.5 Structure of the Thesis

The remaining of the thesis is structured as follows. Chapter 2– Literature review, of the related works and Chapter 3 will present the methodology that we followed to accomplish the research works and Chapter 4 will evaluate the results and outcomes gained from the research, and Chapter 5 will include the conclusion and the future works.

CHAPTER 2

LITERATURE REVIEW

Financial fraud detection has become a highly researched area and also highly subjective to improvements and evolve concerning the growth of its popularity. Fraud detection can be usually seen as a pattern classification problem of identifying normal behavior from abnormal behavior. Different techniques have been used such as data mining techniques and machine learning techniques in developing fraud detection systems. Some of the states of art fraud detection mechanisms used in fraud detection are decision trees, logistic regression, shallow neural networks, support vector machine, and k-nearest neighbor. (Khatri et al., 2020; Kim et

al., 2019; Najadat et al., 2020; Ryman-Tubb et al., 2018). In recent years researchers have applied new approaches alongside different machine learning techniques to improve fraud detection ability. Some of those approaches are briefly explained below.

2.1 Machine learning techniques used in fraud detection

Applying machine learning techniques in fraud detection has become a promising solution in fraud detection. Fraud detection can be identified as a pattern classification problem where to identify normal behavior from abnormal behavior. Different techniques like data mining and machine learning techniques such as supervised learning, unsupervised learning, semi-supervised learning, and Deep learning techniques have been applied in fraud detection systems. Some of the state-of-the-art fraud detection mechanisms have employed logistic regression, shallow neural networks, Support Vector Machines, and k-nearest neighbor (Khatri et al., 2020; Kim et al., 2019; Najadat et al., 2020; Ryman-Tubb et al., 2018). In recent years researchers have applied new approaches alongside different machine learning techniques to improve fraud detection ability. Some of those approaches are briefly explained below.

2.1.1 Supervised learning techniques

Different machine learning techniques have shown effective results in credit card fraud detection. Supervised learning techniques are mainly applied by observing past transactions and combined with rule-based approaches. Most fraud detection research is based on the classification strategy, which mainly consists of labeled data. The main inspiration in supervised learning is to learn from the information in the task, which has been provided in the past(Khatri et al., 2020). Since there exist many types of supervised learning techniques like Nearest Neighbor, Support Vector Machine, Linear regression, Naive Bayes, etc. It's needed to get an idea about the most precise algorithm or the combination of algorithms to detect fraudulent attempts. Random Forest classifier has shown better performances in cases where there exist many input features to learn from. (Nami and Shajari, 2018; Van Vlasselaer et al., 2015; Wedge et al., 2019). Among many supervised techniques, it's shown that SVM is powerful and identified as a universal learning machine utilized in two-class classification problems. (Chen et al., 2018; Kumarage et al., 2019). XgBoost is also a popular machine learning model used in the domains like fraud detection and also it's identified that XgBoost can tackle the class imbalance that creates overfitting (Lei et al., 2020; Priscilla and Prabha, 2020; Zhang et al., 2020)

2.1.2 Unsupervised learning techniques

Unsupervised learning techniques do not require a labeled data set as supervised learning. They can discover unusual behavior by using different clustering mechanisms. They can group different behaviors like customer behaviors, transactional behaviors, or customer behavior into different clusters to identify unusual behavior. The main advantage of using unsupervised techniques is the possibility of finding undiscovered patterns. Even though unsupervised methods can identify new types of fraud their false rate is generally higher than the supervised learning techniques (Delecourt and Guo, 2019).

2.1.3 Semi-supervised learning techniques

This is a combination of supervised and unsupervised algorithms. Furthermore, when fraudsters dynamically change their methods to avoid being detected, the solutions which use traditional fraud detecting tools i.e expert rules, and machine learning methods without adjustments to new fraud attempts are becoming useless(Zhou et al., 2018). Therefore by the combination of supervised and unsupervised learning techniques, it would help to improve the ability of fraud detection.

2.1.4 Deep learning techniques

Deep learning has recently become a highly focused area in Machine learning. Deep learning methods include Convolutional Neural Networks, Deep belief networks, and autoencoders(Zhang et al., 2019). Deep learning is a subset of machine learning techniques that teaches computers to perform tasks that are natural to humans. Deep learning techniques have drawn more attention in detecting fraudulent actions where it can model based on a vast number of feature extractions. By the same, it's shown that in a survey conducted by (Ryman-Tubb et al., 2018) out of the researches done on credit card fraud detection he suggest cognitive computing is a promising research direction and also identified that Expert Systems / Decision trees which are the most established AI techniques used in fraud detection. AI includes case-based reasoning (CBR), Decision Trees (DS).

2.2 Feature selection / Feature engineering techniques.

Feature selection is one of the important techniques that's been used to reduce the feature dimensionally without compromising the performance by the same results of the machine learning models highly dependent on the set of features that's been used because the good

feature variables are proven to perform well and increase the performance of machine learning methods. The research done by Zhang et al (Zhang et al., 2019) proposed a novel feature engineering framework with deep learning models for developing a credit card detection system. This feature engineering framework was based on homogeneity-oriented behavior analysis (HOBA). Two strategies, a transaction aggregator strategy, and a rule-based strategy are applied to fulfill the homogeneity-oriented behavior analysis and extract the feature variables on historical transactional data. They have evaluated the performance of the proposed model using different machine learning techniques and the fraud detection performance is measured by applying both deep learning techniques and traditional machine learning methods. But in their research, they have considered location as a significant factor for fraud detection but when it comes to electronic payment systems' location adds less value because instead of location the IP address is being tracked and if any intruder uses a mechanism like VPN the location tracking adds less value.

Lucas et al and others (Lucas et al., 2020) proposed an automated feature engineering for credit card fraud detection using multi-perspective (Hidden Markov Model) HMM. Their HMM-based approach offers feature engineering to model temporal correlations to improve the effectiveness of the classification. But these HMM-based features cannot be made for the users having fewer transactions and also HMM-based characters cannot characterize some new transactional behaviors of users.

In the research conducted by Ajeet Singh (Singh and Jain, 2019) he proposed several feature selection methods which include the filter method, Wrapper method, and Embedded. The filter method evaluates the data set based on the correlation between the attributes and it’s mainly applied in the data preprocessing phase. The wrapper method is a closed-loop method to evaluate the problem based on the learning algorithms of subset evaluation and the embedded method is a combination of both filter and wrapper method.

(Wedge et al., 2019)proposed a method to overcome the false positive declines in fraud prediction using automated feature engineering technique and they have adopted a Deep Feature Synthesis technique to automatically derive behavioral features based on a historical transactional data set associated with card payment. There he has used the feature tools to generate the relationships and he hasn’t considered the RFM features which can generate rich features alongside the Deep Feature Synthesizing techniques they also have used a banker level transactional data set.

In (Van Vlasselaer et al., 2015)’s APATE method he has proposed a network-based extension for credit card fraud detection for his research he has used a one-month banker data set and has

shown that features generated with APATE method perform well with the Random Forest classifier. He also mentioned that the feature engineering technique that he used fits the popular RFM framework. The drawback of Van Vlasser's APATE model is that he hasn't addressed the problem of false-positive declines and by the same one-month data set is not enough to conduct a behavior analysis.

(Xie et al., 2019)Proposed a feature extraction method based on the frequency and rule-based approaches to minimize credit card frauds but in his research, he hasn't considered the recurrency and the monetary value which depicts interesting information about the user's behavior.

2.3 Summary

When considering our identified problem and the past research that has been conducted, the most appropriate approach is the application of feature engineering techniques to summarize the user's behaviors. But we have identified some drawbacks of the currently used approaches. Some of them are, almost all of the researchers have put their main focus on fraud detection and prevention and the problem of raising FP problem has gone unnoticed. The raise of FP's is mainly affected by merchants and other electronic payment solution providers. By the same, those researches have been done based on transactional collected from banks and Kaggle data sets. The Kaggle datasets provide less information and they don't consist of much historical data related to the transactions and also some of them include synthesized data and the feature variables are anonymized in those datasets due to security constraints, therefore, using Kaggle datasets for behavior analysis can produce false predictions and erroneous results.

The banker data sets have a rich data set but they have to address multiple dimensions where they have a collection of credit card transactional data set consists of transactions which includes the card-present transactions as well as card, not present transactions and also they includes all kinds of transactions like cash withdrawals, payments made to terminals. To the best of my knowledge, none of the researchers have used an application-level transactional data set containing only CNP transactional data.

Other than the above-mentioned problems it can be seen that almost all of those researches haven't focused on the feature engineering techniques and behavior analysis techniques that can be used with Application-level data sets that contain only CNP transactions.

2.4 Research Gaps

The following (Table 1) summarizes the research gaps identified in the literature review based on the different feature engineering techniques.

Author & topic	Data set used	Findings	Limitations / Gaps
HOBA: A novel feature engineering methodology for credit card fraud detection with deep learning architecture (Zhang et al., 2019)	Bank card transactional data set	Use location as a significant fact to identify fraudulent behavior RFM based features can be used to depict user's transactional behavior HOBA with DBN perform well than RFM	Location-based feature engineering doesn't add much value to electronic payments Research is done in the presence of card-present as well as card, not present transactions Banker level transaction data set being used Do not address the problem of false-positive declines
HMM-based feature engineering for credit card fraud detection (Lucas et al., 2020)	Real-world transactional dataset provided by Europian card processing company	HMM provides better feature variables	HMM-based features cannot be calculated for the users having less no of transactions HMM features cannot characterize new transactional behaviors of users
APATE: A novel Approach for credit card fraud detection using Network-Based extension (Van Vlasselaer et al., 2015)	Used a one-month data set from the European card issuer.	RFM based features combined with network-based features leads to provide higher AUC score	Research is done in the presence of card-present as well as card, not present transactions Banker level transaction data set being used Do not address the problem of false-positive declines

			Rule-based approaches are not used
Adaptive credit card fraud detection technique based on feature selection (Singh and Jain, 2019)	German credit dataset is used	Filter and wrapper helpful in selecting high correlated features to increase the efficiency of machine learning techniques	Research is done in the presence of card-present as well as card, not present transactions Banker level transaction data set being used Do not address the problem of false-positive declines Rule-based approaches are not used
Solving the false positive problem in fraud detection (Wedge et al., 2019)	Data set collected from a large multinational bank is used	A rich data set is used User Deep feature synthesizing techniques Use feature tools to generate features	Research is done in the presence of card-present as well as card, not present transactions Banker level transaction data set being used Rule-based approaches are not used
A feature extraction method for credit card fraud detection (Xie et al., 2019)	Considered a transactional data set from a financial company in china	Considered frequency-based approaches as well as rule-based approaches	Research is done in the presence of card-present as well as card, not present transactions Do not address the problem of false-positive declines Recurrent behavior is not taken into account.

Table 1: Research Gaps

CHAPTER 3

METHODOLOGY

3.1 Data set preparation

3.1.1 Overview of DirectPay data set

The transactional data set is based on a real-life transaction data set obtained from DirectPay, which is one of the leading electronic payment solution providers in Sri Lanka. In the initial phase, the transaction data set consists of more than 800,000 transaction records, and out of them, we have retrieved a data set belonging to the period between 2020-06-01 to 2021-05-01. Which included nearly 500000 transactions. We have considered transactions related to a period of one year because in our research we mainly focus on the transactions committed within the three aggregation periods which include yearly, monthly, and weekly. This transaction data set mainly consists of transactions made via mobile wallets, IPG (Internet Payment Gateway), and mPos (Mobile Point Of Sale).

Mobile wallets allow the users to link their credit/debit cards and accounts to the wallet in advance to make transactions and payments. Mobile wallets' popularity has been raised especially during this pandemic season where people are much more interested in making contactless payments to avoid the deadly virus. In the same sense, with the increase in the popularity of mobile wallets, it’s becoming a prominent target of fraudsters.

Internet Payment Gateway or IPG is a merchant service provided by Directpay which accepts electronic payments which include credit/debit card payments and bank transfers this has been provided as Software As A Service (SAAS) model for merchants. This helps to integrate multiple payment methods to their websites, e-business and collect payments easily.

mPos or mobile point of sale solution is also a service provided for the merchants which enables them to collect payments by sending a payment link to their customers where the customers can easily make transactions by entering their card details to the link provided. Apart from that bill payments and QR payments are enabled providing merchants a new experience and convenient method to collect their payments.

3.1.2 Payment process

Figure 1, clearly shows how these electronic payment methods are implemented and how they operate. When a transaction is made using a Card/Account through a mobile wallet, Web (e-commerce site), or a mPOS all those transactions can be considered Card Not Present

transactions, where the user has to enter only the card details to proceed with the payment. Once a payment request is made It's been redirected to the payment processor. Payment processor consists of acquirer bank, issuer bank, and MPGS gateway. The bank is responsible for processing the payment, If the transaction request is accepted it will credit the merchant's account and debit the card/account holder's account and send a "Success" response. In a case where the transaction is failed due to some reason, there can be many reasons. Examples of some instances where the transactions are rejected are the user has entered the card/account number incorrectly, has entered the CVV number incorrectly, has insufficient balance, etc. For all these kinds of instances, the bank will reject the transaction and provide a response code with generalized reasoning. In most occurrences, the bank does not disclose the actual reason for rejection and In instances where the transactions are declined by their fraud detections systems, it gives a meaningless error response like 'Do not honor' which doesn't give the actual reason. Sometimes those transactions are rejected due to some suspicious behavior, Banks enclosed the actual reason for rejection to avoid identifying their fraud detection mechanism by an intruder. From this, we can see that an application-level transactional data set contains one-time filtered data set from the Banker's fraud detection system so their main concern is minimizing possible FP declines while maintaining a lesser FN ratio.

3.1.3 Data cleaning and preprocessing

We have used one-year transactional data set collected from Directpay in our research. In the initial cleaning process, we have categorized this transaction data set into three major subcategories such as.

1. IPG transactions
2. Mobile wallet transactions
3. mPos transactions

When considering the facts like transaction volume, failure rate, no of "Do not honor" responses (Table 2) We can see that IPG transactions have shown high transaction volume and at the same time in IPG transactions the failure rate, and No of "Do not honor" responses are also higher with related to Mobile wallet transactions and mPos transactions. Therefore taking into account the critical nature and the vulnerability of IPG transactions we managed to conduct the research based on the IPG transaction data set.

Payment Category	Transaction Volume	Failure rate	Do not honor responses
Mobile wallet	73, 030	13%	<30

mPos transactions	163,107	8%	<100
IPG transactions	256,284	56%	>20000

Table 2: Payment Categories

Altogether we have collected 256,284 transaction records including 249,579 normal transactions and 705 fraudulent transactions. The total aggregated loss caused due to these fraudulent transactions are exceeding 2 million rupees. When it considers the percentage of actually incurred fraudulent transactions it's very low but the loss incurred and the damage is very high because once a user has gone through a bad experience they usually share their experiences with social media and other platforms giving a bad reputation to the company or the financial organization. Apart from these fraudulent transactions when it considers the banker responses 23672 transactions have been rejected due to "Do not honor response" which involves nearly 10% of the total transactions. The aggregated value of those "Do not honor" transactions exceeds 230 million rupees. These "Do not honor" responses from banker occur mainly when their fraud detection mechanisms declined those transactions and sometimes due to the identification of some anomalous behavior of the user. When we consider the labeled fraudulent transactions it can be seen that there exist some transactions that the bank has approved and later on identified as fraudulent transactions. This provides us an important insight that there can be false declines among those "Do not honor" declines and at the same time there's a possibility that a bank-approved transaction can become a fraudulent transaction.

To answer our main research question, how to minimize the false-positive declines in electronic payment systems study of the past transactional behavior of users is very important. Past transactions are the best interpretation of customer behaviors in the system. Feature engineering techniques are expected to construct feature variables that properly summarize the transaction behaviors of individuals based on the raw transactional records

In the first phase, we had to go through some clearance procedures and obtain the necessary permission to conduct research based on the DirectPay data set since it consists of some sensitive data like user details, account masks, transaction values, etc. we had to follow some standard techniques in data acquiring process to obtain data without affecting the sensitivity of the data set. The DirectPay data set consists of hundreds of feature attributes and in the first phase, we needed to identify the most effective attributes for feature engineering techniques. Applying domain knowledge and after analyzing the data set we have selected 18 feature attributes and (Table 3) summarizes the initially used feature attributes.

Lable encoding - In the first phase, we had to follow some synthesizing techniques in the conversion of the data set. The data set consists of some sensitive information related to the

users and transactions like payer details, merchant information, and card data which expands over a larger dimension are synthesized using label encoding techniques to convert those string type data to numerical form to feed to machine learning models.

Specialization – Some fields we had to expand for eg. The time field consists of a timestamp and we had to generate two fields separately as date and time from the timestamp to make the analysis process more convenient.

Generalization – In the cleaning process some categories are generalized to a common categorical type for eg. Banker responses received for time out, technical errors, missing parameters are categorized into a general class as errors. This categorization is conducted by considering the similarity of the characteristics they share.

One hot encoding – Here we have used one-hot encoding for categorical data. Our data set consists of many categorical values like status, type of the transaction, banker responses, currency, etc., and these categorical values have to be identified clearly when we proceed with the feature engineering. For Eg. We need to identify how many successful transactions have been conducted by the user for that purpose we consider the transaction status.

Rule-based approaches – Apart from the above-mentioned approaches we have used some rule-based approaches to summarize the user's transactional behavior. These rule-based approaches have been taken to make the feature engineering process easier and to derive complex rules based on the derived rules. Those complex rules have been briefly described in chapter 3.2.1.

After applying all of those techniques for the initially obtained 16 feature attributes we have ended up with 34 feature attributes.

Level of information	**Attribute**	**Description**
User	Unique user Id	Used to identify the person who made the transaction.
	Card Mask	The card used to make the transaction
Merchant	Unique merchant Id	Identity of the person who receives money
	Business type	Type of business of the merchant i.e fashion, retail, e-commerce
Transaction	Status	Status of the transaction, whether a success or a failure
	Amount	Transaction value

	Transaction type	Type of the transaction which a one-time payment or a recurrent payment
	Banker response received	Acquirer bank response received for the transaction. 1. Total bank approved transactions 2. Total Do not honor responses 3. Total insufficient balance responses 4. Total bank error responses 5. Total restricted / lot card responses 6. Total bank invalid card responses
	Currency	Type of the currency used for the transaction i.e 'LKR' or 'USD;
	Time	Date and time which the transaction occurred
	Card type	Type of the card i.e VISA, MASTER, AMEX
	Issuer Bank	Card issuer bank
	IP address	The IP address of the device used for the transaction
	Browser	Type of web browser used to make the payment
	Funding method	Whether the card used for the transaction is a Credit card or a Debit card
	Secure 3ds	Whether the card has enabled 3ds authentication
	Currency	LKR transactions or a foreign currency transaction
	Transaction initiating channel	Channel that used to commit the transaction

Table 3: Features obtained after pre-processing

3.1.4 The behavior of preprocessed data set in the presence of machine learning models

Before going on to the application of further feature engineering and deriving new features, first of all we needed to evaluate the performance of the original data set after applying the basic filter and cleaning approaches. According to the literature review that we have conducted

we came across that Random Forest classifier and XGBoost classifier are some of the supervised learning techniques that have shown better results in fraud detection classification problems. If we consider our data set it's highly imbalanced and when we feed data to the classifiers we have experienced model overfitting because the labeled fraudulent transaction data set belongs to a single merchant. Then we applied some techniques like undersampling the major class and oversampling the minor class by synthetically generating fraudulent transactions. For synthetically generating the fraudulent transactions we have applied the domain knowledge and the previous experiences that we had with fraudulent transactions in this manner we have generated 2000 more transactions synthetically. We haven't chosen SMOTE as our oversampling technique because we have experienced that SMOTE works by generating new instances from the existing minority cases that we have supplied as input such that we have experienced it doesn't add any value to improve the learning process of the classifier. After synthetically generating the fraudulent transactions and oversampling the minority class we have applied the Random Forest and XGBoost classifiers to our DirectPay data set to observe the behavior(Table 3). We can see that the Random Forest classifier and XGBoost classifier results respectively.

Algorithm	Precision	Recall	F1-score	AUC
Random Forest	0.74	0.35	0.48	0.956
XGBoost	0.81	0.32	0.46	0.990

Table 4: Classification report obtained for the initial data set

When it considers the fields used by the two algorithms for model training (Gini Index) we have found that it's not the same fields that are been used by Random Forest Classifier and XgBoost classifier for the model training. Taking into account the fields used by both classification models and applying domain knowledge we have finally come up with 26 feature attributes to be used in applying feature engineering techniques. The main objective of feature engineering is finding the best features that can summarize the user's transactional behavior such that those features can affect the learning process of the classifiers.

3.2 Feature engineering framework based on behavior analysis

To answer our main research question, how to minimize the false-positive declines in electronic payment systems study of the past transactional behavior of users is very important. Past transactions are the best interpretation of customer behaviors in the system. Feature engineering techniques are expected to construct feature variables that properly summarize the transaction

behaviors of individuals based on the raw transactional records. (Table 5) Summarizes the data set that we have obtained after data processing and cleaning approaches which we have described in the previous section.

Attribute	**Description**
Payer_account_number	Card mask used for the transaction
Payer_id	User id to identify the user who commits the transaction
Payee_id	Marchant id to identify the person who receives money
Original_amount	Original transaction amount
Created _at	timestamp
Transaction_date	Date of the transaction
Transaction_time	Time of the transaction
Isuuer_bank	Card issuer bank
Ip_address	Ip address of the device
browser	Device id of the device use for the transaction
type	Card type credit/debit card
Secure_3ds	3ds enabled cards
Success_3ds	Successfully authenticated 3ds authentication
status	Status of the transaction success/failed
currency	LKR or a USD transaction
Amount_high	Transaction amount >10000
Suspicious_time	Transaction done on non working hour
High_risk_credit_transaction	Credit card transaction amount >10000
Bank_approved	Transaction approved by bank
Bank_error	Transaction rejected due to bank error
Bank_invalid_card	Transaction rejected due to invalid card credentials
Bank_insuficient_fund	Transaction reject due to insufficient balance
Bank_do_not_honor	Transaction reject suspecting for fraud

Bank_lost_card	Transaction attept using lost or canceled cards
Is_fraud	Identified fraudulent transactions

Table 5:Selected features to apply feature engineering techniques

According to the Literature review (Van Vlasselaer et al., 2015) proposed that customer spending habits using the fundamentals of RFM (Recurrent, Frequent and Monetary framework) can be used for credit card fraud detection. RFM framework is a widely used marketing analysis tool to identify a firm's best clients according to their spending habits. But we have found out that RFM framework is not a proper mechanism to conduct a behavior analysis for electronic CNP transactions. For e.g As it's shown in (Table 6), we can see user 1 has conducted 447 transactions with a total aggregated value of Rs 1,378,967 using the card 1212121 while the most recent transaction is conducted a day ago. Considering only the RFM features and calculating the RFM score this customer can be categorized as a good customer (We have calculated the score). But it can be seen that only a single transaction valued Rs. 3147 has become a success out of 447 transactions which implies very suspicious behavior.

Account Number	Payer Id	recency	frequency	Total value	Total success amount	Success count
1212121	1	1	447	1378967	3147	1

Table 6:Application of RFM in CNP transactions

Furthermore, Since our major concern is addressing the false positive decline issue we took into consideration the banker responses received for a particular transaction. For e.g For a particular card user having a high failure rate due to "Do not honor" transactions are we have to pay extra attention to such transactions even though we receive an "Accept" response from the banker side. And the other thing around which is if a particular user has shown good transaction history with a high success rate if such a customer receives a "Do not honor" such kind of transaction is more likely to be a false decline. This is an important fact that needed to be considered so we use the characteristics like the success rate, failure rate, banker responses, alongside the RFM features. The basic RFM framework which used in marketing analysis doesn't account for such characteristics in their model.

Therefore in our proposed model we mainly focus on the intrinsic and homogeneity behavior of CNP electronic payments. In our proposed model we first need to identify the set of

characteristics that summarizes the customer's behavior. For that purpose, the behavior analysis is mainly carried out considering the Recency, Frequency, and Monetary values.

Recency – How recently a customer has made a transaction with the given set of characteristics

Frequency – How frequently/often a customer has made a transaction with a given set of characteristics

Monetary value – How much did a customer has spent on a transaction with a given set of characteristics.

To generate rich and complex features base on the historical transactions we have adopted transaction aggregations strategies and rule-based approaches. According to the literature, we follow the four main aggregation strategies which include the aggregation characteristics, aggregation period, transaction behavior measure, and aggregation strategies. The approaches that we have used to generate the characteristics used for RMF features are described in the following section. The features are generated considering the user and the card they used to conduct the transaction.

3.2.1 Approaches used for feature generation

- **Aggregation strategies**

Under aggregation strategies, we have applied the aggregation strategies like the count, sum and generated the new features considering the period. Eg. Tot no of Do not honer responses received for a particular user for a particular card within the given aggregation period. Appendix I has attached the derived features and the functions used to generate the features.

- **Aggregation characteristics**

Under aggregation characteristics, we have considered the derived features like average, mean, standard deviation, etc. We use these aggregation strategies to derive the behavioral characteristics out of transaction records. Eg. average transaction amount, average decline rate, etc.

- **Transactional behavior measure**

The transactional behavior of each individual is been summarized by using rule-based approaches. All these rules are end up with a Boolean value where it matches the given rule it returns one or zero otherwise.

Rule 1: *Suspicious frequent transaction attempts per day* – A fraudster can attempt transactions at different times of the day, to check whether the card is working or not and can commit transactions having a lesser time gap in between the transactions. And also they can attempt several times. To identify this kind of behavior it can define the maximum allowed transaction attempt count and also should define the time gap related to the transaction volume. If this kind of suspicious behavior is identified a new feature "suspicious_transaction_attepts_perday" is added and marked with a 1, otherwise 0.

T_d - The time difference between two consecutive transactions

T_{max}- Max allowed transaction attempts per day

T_{count}- Transaction count

$$= \begin{cases} 1 & T_d < T_{threshold} \ and \ T_{count} > T_{max} \\ 0 & Otherwise \end{cases}$$

Account number	Payer Id	Date	Time	Amount
1212121	1	2020-08-10	02:56:56	1877
1212121	1	2020-08-10	03:01:08	1825
1212121	1	2020-08-10	03:06:22	6507
1212121	1	2020-08-10	03:06:28	1800

Table 7:Example for Rule 1

Rule 2: *Suspicious high amount transaction attempt:* Amount can be considered as an important feature in making decisions. If a fraudster gets a card his main intention is to get all that it left. So within a short period, there's a possibility of attempting large valued transactions. this kind of behavior can be identified by the rule given below. If this kind of behavior is identified "suspicious_per_day_high_amount" is marked as 1, otherwise zero

T_d - The time difference between two consecutive transactions

a – Transaction amount

A_{large} - Large transaction amount and this is defined by the system. (In our research we have considered the JustPay limit as the large value transaction amount). This amount needed to be defined by the system according to their transactional volumes.

$T_{threshold}$ – Time threshold is defined by the system (The time gap)

$$= \begin{Bmatrix} 1 & T_d < T_{threshold} \ and \ a \in A_{large} \\ 0 & Otherwise \end{Bmatrix}$$

Account number	Payer Id	Datetime	Amount
1212121	1	2020-06-03 21:28:56	55528
1212121	1	2020-06-06 21:33:51	25000

Table 8:Example for Rule 2

Rule 3: *Suspicious payday max limit:* A fraudster can apply different techniques in the same way of attempting the opposite also can happen. They can attempt transactions with small amounts to avoid getting highlighted if it is observed even the one-time transaction amount accounts for a low value the aggregate sum can accumulate to a larger value. If this kind of behavior is identified "suspicious_per_day_high_sum" feature is scored with 1, otherwise 0.

n – represents the size of the group of small transactions.

i and n represents integers respectively.

T_d - The time difference between two consecutive transactions

a – Transaction amount

$$= \begin{Bmatrix} 1 & T_d < T_{threshold} \ and \ \sum_{i=1}^{n} a_i \in A_{large} \\ 0 & Otherwise \end{Bmatrix}$$

Rule 4: *Suspicious test transaction:* Usually when a fraudster receives a card they attempt to make sure the card is working so they usually try with a small amount before trying a large transaction and commit a large transaction in a case where the small amount gets success. If this kind of behavior is identified "suspicious_test_transaction" feature is marked with 1 or 0 otherwise.

$T_d = (T_{i+1} - T_i)$ - The time difference between two consecutive transactions where

a – Transaction amount

A_{small}- Small amount (according to our analysis we have considered 1000 as a small value threshold) Currently we have considered these amounts according to the Payment and Settlement Department report of the Central Bank.

A_{large} – Large valued amount (According to our data set we have considered 10000 as a large amount threshold which is the JustPay limit)

$$= \begin{cases} 1 & T_d < T_{threshold} \ and \ (a_{i+1} \in A_{small} \cap a_i \in A_{large}) \\ 0 & Otherwise \end{cases}$$

Account Number	Payer Id	Datetime	Amount
1212121	1	2020-06-03 05:40:56	10
1212121	1	2020-06-03 05:44:05	70000

Table 9:Example for rule 4

Rule 5: *Suspicious failure rate rule:* When a fraudster tries to do a fraudulent transaction they attempt those transactions applying different techniques and other than that they regularly change their pattern. Even though their transactions accidentally get success their success rate lays at a very low level. To identify such a scenario we consider the failure rate if the failure rate is > 50 % we consider it as suspicious behavior. Here we have defined 50% as the failure rate threshold by analyzing our data set this value has to be set after thoroughly analyzing the data set. If it identified a suspicious failure rate then the "suspicious_time_transaction" feature is scored with 1 or 0 otherwise.

T_{fail}- Transaction failure rate

$$= \begin{cases} 1 & T_{fail} > 50\% \\ 0 & Otherwise \end{cases}$$

Account Number	Payer Id	Frequency	Success count	Failure rate
1212121	1	301	0	100%

Table 10:Exapmle for Rule 5

Rule 6: *Suspicious banker response*: Our transaction set consists of a one-time filtered transactional data set. Therefore the banker responses provide better insights into the behavior of the transaction. Suspicious banker score also we have considered it is two-fold.

1. Suspicious one time – If the transaction is labeled as a one-time payment and its failed ratio is higher than 50% due to a suspicious bank response then we considered such behavior as anomalous behavior. We have generalized banker responses into six main categories and other than "bank_accept" all the other categories are considered as suspicious banker responses for a one-time transaction.
2. Suspicious recurring payment – We have observed that there's a high probability that recurring payments can get failed due to "Insufficient balance". Therefore for recurring payments, if the transactions get failed only due to insufficient balance then we don't consider it as high risk but if recurring payments get failed due to other banker responses and the failure rate due to other reasons exceeds 50% then we consider it as an anomalous behavior

If the transaction failed due to a suspicious banker response "suspicious_banker_response" field mard with 1 or 0 otherwise.

T_{one} – One-time payment

$\overline{T_{one}}$ – Recurrent payment

a – transaction

T_{fail} – Transaction failure rate

T_{insuf_bal} – Failure rate due to insufficient balance

B_{res} – Suspicious banker response

$$\begin{cases} 1\ (a \in T_{one}\ \Lambda\ a\ \in\ B_{res})\ and\ T_{fail} > 50\%\ \cup \left(a\ \in\ \overline{T_{one}}\ \Lambda\ a\ \in \left(B_{res} - T_{insuf\_bal}\right)\right)\ and\ T_{fail} >\ 50\% \\ 0\ \ Otherwise \end{cases}$$

Account number	Payer Id	Frequency	bank_approves	bank_errors	invalid_card	Insufficient_fund	Lost_card	Do_not_honor
1212121	1	52	2	0	0	0	31	19

Table 11:Example for Rule 6

Rule 7: *Suspicious transaction time rule:* Normally transactions done within unusual times are considered as suspicious behavior such that one-time transactions done in the period of (1.00 AM – 5 AM) are considered as auspicious time transactions. If a transaction is committed within an unusual time "suspicious_time_transaction" field scores with 1, 0 otherwise.

a –Transaction

N_{hour} – Non-working hour

$$= \begin{cases} 1 & (a \in N_{hour}) \\ 0 & Otherwise \end{cases}$$

Account number	Payer Id	Date	Time
1212121	1	2020-06-01	01:44:45
1212121	1	2020-06-01	02:10:33
1212121	1	2020-06-01	03:03:50

Table 12: Example for Rule 7

Rule 8: *Suspicious credit transaction:* Under this rule, we consider suspicious credit card transactions conducted by the user. Credit card transactions are riskier than debit card transactions due to chargebacks. If a user attempted high valued credit transaction from a card where 3ds secure is not enabled "suspicious_credit_transaction" field scores as 1 or 0 otherwise.

A_{large} – Large transaction amount this amount is defined by the system

a_c – Credit card transaction

$3d$ -3D secured

$$= \begin{cases} 1 & a_c \; and \; a_c \in A_{large} \; and \; \overline{3d} \\ 0 & Otherwise \end{cases}$$

Account number	Payer Id	Secure_3ds	Amount	Credit_card
1212121	1	0	64723	1

Table 13:Example for Rule 8

Rule 9: *3DS Authentication failed suspiciously:* 3DS security is used to authenticate the user. In some cards, it allows to do transactions without 3DS authentication and in some cases, it's found that even the 3DS authentication is enabled the 3DS verification also failed suspiciously. These kinds of behaviors are identified by this rule.

$$= \begin{cases} 1 & 3D \; secured \; \Lambda \; 3DS \; failed \; V \; 3DS \; bypassed \\ 0 & Otherwise \end{cases}$$

Rule 10: *Suspicious card usage:* Fraudsters can try transactions impersonating different users. Eg. Use the same card details with different user details like using different phone numbers and a different username and those data have been changed frequently. To identify this kind of suspicious behavior we have considered the number of distinct users who have used the same card. If the same card is used by more than three users we consider it as suspicious behavior. If this kind of behavior is identified "suspicious_card_usage" field is scored with 1 or 0 otherwise

- **Aggregation period**

Under the aggregation period, we mainly consider three aggregation periods last week, last month, and last year such that all the generated features and customer transactions are aggregated with related to all of those aggregation periods. Appendix A shows the sample characteristics that we have obtained for the three aggregation periods.

3.3 Identifying best customers of electronic CNP payment solution providers using a scoring index

Based on the features generated rule-based features which we have described under transactional behavior measures can be used to summarize the user's behavior in a CNP electronic payment system environment. Those generated rules end up with a boolean value where if the condition matches it will return a 1 either 0. Those generated rules can be applied threefold. The three folds include yearly, monthly, and weekly scores. These scores are generated by observing the past behavior of the users.

3.3.1 How the score is generated

We have set a threshold transaction count as ten where we generated the past transaction-based score only for the users who have committed transactions above the threshold. By following this mechanism we can generate a score out of thirty for each user. If any user has a transaction history of fewer than 10 transactions we start their scoring index from thirty (Rules are been generated three-fold and each fold gets a score of 10 and therefore the user will receive a score out of 30 (10*3)). For eg., We consider the transaction frequency for each aggregation period, if any user conducted transactions only during the last week and there is no transaction history recorded for the aggregation period last month and last year then for those two aggregation

periods the user will receive the maximum score assigned for each period which is 10. If the weekly transaction count is greater than the transaction threshold then the weekly score will be generated based on the rule-based approaches such that the user can receive a score in between [20-30].

Here it has described briefly how those rules are been generated.

$R = \{r_1, r_2, \dots\dots\dots\dots\dots, r_{10}\}$ Where R represents the generated Ten rules. As mentioned earlier those rules are been generated in three folds such as Yearly, Monthly, and Weekly

R_{Yearly} – Rules generated considering yearly transactions

$R_{Monthly}$ – Rules generated considering monthly transactions

R_{Weekly} – Rules generated considering weekly transactions.

$$Critical\ Index = \sum_{i=1}^{10} R_{yearly} + \sum_{i=1}^{10} R_{Monthly} + \sum_{i=1}^{10} R_{Weekly}$$

Here we have given equal weights because we don't have any previous study results since it's the first time it's been suggesting this kind of scoring mechanism we have given equal weights for all the rules based on the assumption that there's equal likely opportunity to occur any of this kind of behavior summarized by the rules. By observing the results and the progress of our proposed model we plan to optimize this scoring index.

According to this scoring index, those who obtain higher values can be considered as the customers who have a bad transaction history or either they have very little transaction volume. Once we generate the scores we have observed the distribution of the scores. Figure 7, Figure 8, and Figure 9 show the distribution of yearly, monthly, and weekly scores generated. In figure 10 It shows the distribution of the final critical index generated. Base on the final critical index we can categorize our customers mainly into three clusters. Table 16 shows the three clusters that we have identified according to the distribution. Customers who have been categorized as low risk are frequent customers and have a clean transaction history. The probability of getting fraud from that customer category is low. The user category which has been identified as moderate has a relatively good transaction history while those who have identified under the high-risk category have either a less transaction history or have a suspicious transaction behavior. The main purpose of this scoring index is to identify the best customers.

3.4 Proposed Model/Framework to minimize FP declines

In this section, we have described the proposed framework based on our proposed feature engineering technique. The fraud detection and the decision-making process are described in the figure.

As it is shown in this figure when a CNP transaction request is been made via a mobile, web, or mPos it's been redirected to DirectPay middleware. From the DirectPay middleware, the transaction request is been recorded in the database and sent to the Acquirer bank. The acquirer bank is responsible for committing the transaction which includes crediting the merchant's account and debiting the payer's account, while this process is been going on the transaction request is been directed to the fraud detection module (Application-level module). First, it calculates the critical index for the user based on the scoring index that we have proposed. Through this score, we can identify whether the transaction is committed by low risk, high risk, or a moderate customer, and at the same time, the corresponding suspicious score is also calculated based on the prediction given out by the classifier. The training of the model solely depends on the historical transactional records of the user using the corresponding card and we consider only one-year historical records. In the next step, the transaction, banker response received for the transaction, critical index, and the suspicious score generated by the training model are been set as inputs to the decision-making module.

The decision-making module first considers the customer category and the banker responses. The banker response indicates whether the transaction is a success or declined from the banker side and also we consider the prediction given out by our training model. The prediction is given out by the score generated by the training model and this cutoff score is determined according to the tolerance of the false positive rate(FPR) of the training model. If the banker response is a success and if the customer belongs to the high-risk category we consider the prediction given out by our model if it was also lower than the cutoff we accept the transaction and in a situation where the DirectPay prediction gives a high score for a high-risk category customer even the transaction is accepted by the banker side, we issue an alert even the transaction gets approved by the bank still there's a possibility of fraudulent transactions this kind of transactions can be filtered through this decision. In the same way, if the customer belongs to high risk or moderate category banker response is a success, and the DirectPay model prediction score is also low then the transaction is accepted. In a case where the

DirectPay model prediction gets a higher score for a low or moderate customer, the transaction is reviewed even it’s a success from the banker side.

The other scenario is from the banker side the transaction gets declined in such a condition if the customer category is high risk and the prediction given out from the DirectPay model is scored as high risk then the transaction is rejected by giving a risk alert. If the Prediction of DerectPay model is low risk but the customer category belongs to the high-risk category and in that scenario also it issues an Alert. In the same way, if the considering customer category is identified as low risk or as moderate and the prediction given out from the DirectPay model is low risk in such condition if the bank transaction is gets declined we review the transaction to avoid false-positive decline. If the DirectPay model prediction is also high risk and the transaction gets declined from the banker side even the customer category belongs to the low or moderate category an alert is issued. This is done to identify any anomalies behaviors and to avoid fraudulent transactions targeting legitimate customers.

In the final stage, the decisions obtained from the decision-making module are sent for review and alert management system. There they investigate the transactions and make the final decision whether it’s a legitimate transaction or not and update the status of the transaction in the database. In this paper, we have mainly focused on an offline training model. The model is feed with the features generated by applying the feature engineering techniques that we have described in the previous section.

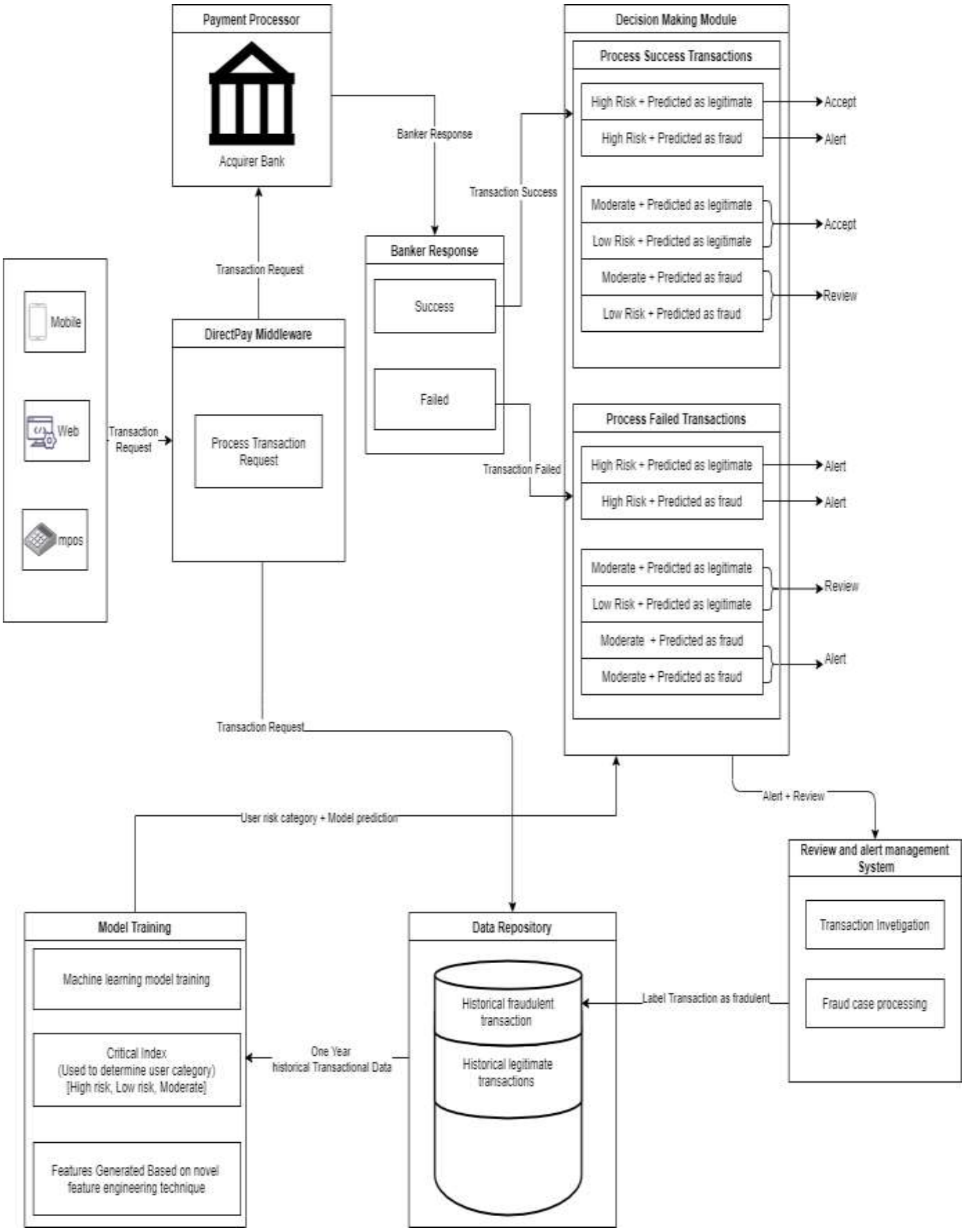


Figure 1:Proposed Model

CHAPTER 4

EVALUATION AND RESULTS

This chapter has briefly described the evaluation process and the results derived throughout the study.

To evaluate our model we use the classification report results obtained for the initial data set (Transactional features) obtained after the cleaning process as our baselines.

In the baseline, we use only the features generated at the time of the transaction. We do not include any new feature generated using RFM features in the initial phase we have used some techniques like specialization, generalization, One-Hot Encoding for categorical data, and some rule-based approaches (described under the data cleaning process). and altogether generated a total of 33 features.

4.1 Results

Concerning the literature review, we have identified that supervised learning techniques have shown better performances based on the past historical transactional data sets. Among the widely used techniques Random Forest Classifier, XGBoot classifier, and SVM has shown better performances. Concerning the DirectPay data set, we have observed the results obtained for the above-mentioned three classification models.

4.1.1 Predicting results obtained for the original data set

Initially, all three models were trained using the DirectPay data set. The data set consists of 256,284 transaction records including 249,579 normal transactions and as mentioned in Section 3.1.4 to address the problem of class imbalance we have synthetically generated 2000 more fraudulent transactions altogether making 2705 fraudulent transactions and still, it's highly imbalanced but added more values to improve the minority class of the classifier. The original data set is then divided into two as training and testing in the proportion of 70% for the training data set and 30% for the testing data set. The results obtained for the data set distribution are based on the classes "0" and "1". In the training data set there exist 211 transactions labeled as fraudulent (classified as 1) and 74873 transactions labeled as legitimate transactions (classified as 0).

All the classifiers were trained based on the k-fold cross-validation in k-fold cross validation k defines the number of folds in which to split the data set. Here the k is set to10 and used the

10 fold cross-validation. In 10 fold cross-validation, 10 equal-sized subsets of training and validation sets are randomly generated. Then 90% of the data is used for training and 10% is used for validating the model and this procedure is repeated 10 times with each subset 90% for training and 10% for testing. This approach is used for model evaluation and model training to avoid the overfitting of the classifiers.

The results obtained for each classifier after 10 fold cross-validation are given below

4.1.2 Predicting results obtained for the initial data set

This section, it describes the classification reports obtained for the selected classifiers using the initial data set. When it studies the classification reports we can see that the Random Forest classifier and the XgBoost classifier have shown better performances. Considering this behavior we only consider the Random Forest classifier and the XgBosst classifier in our evaluation processors.

- **Classification report obtained for Random Forest classifier**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76691
1	0.71	0.37	0.49	198
accuracy			1.00	76889
macro avg	0.85	0.68	0.74	76889
weighted avg	1.00	1.00	1.00	76889

Figure 2:RF classification Report

- **Classification report obtained for the XgBoost classifier**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76691
1	0.67	0.31	0.43	198
accuracy			1.00	76889
macro avg	0.84	0.66	0.71	76889
weighted avg	1.00	1.00	1.00	76889

Figure 3: XgBoost Classification Report

4.1.3 Predicting results obtained for the data set after applying feature engineering techniques

This section describes the results obtained for the Random Forest Classifier and the XgBoost classifier in the presence of the data set obtained after applying feature engineering techniques.

- **Classification report obtained for Random Forest Classifier**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76683
1	0.64	0.47	0.54	206
accuracy			1.00	76889
macro avg	0.82	0.74	0.77	76889
weighted avg	1.00	1.00	1.00	76889

Figure 4: RF classification report after applying feature engineering

- **Classification report obtained for XgBoost classifier**

	precision	recall	f1-score	support
0	1.00	1.00	1.00	76683
1	0.69	0.34	0.46	206
accuracy			1.00	76889
macro avg	0.85	0.67	0.73	76889
weighted avg	1.00	1.00	1.00	76889

Figure 5: XgBoost classification report after applying feature engineering

4.2 Evaluation metrics

To evaluate the performance of the proposed model and the generated features we have employed some supervised learning techniques. The commonly used performance measure matrices based on the confusion matrix are employed in this paper to evaluate the fraud detection performances of the proposed model. These include Precision, Recall, F-Score, and the Area Under Curve (AUC-ROC). Since our data set is highly imbalanced we mainly focus on the F-Score and the ROC Curve.

Precision – Is the fraction of the true frauds among all samples which are classified as frauds

$$p = \frac{TP}{TP+FP}$$

Recall – (Also known as sensitivity) Is the fraction of frauds that have been classified correctly over the total amount of frauds.

$$R = \frac{TP}{TP+FN}$$

AUC-ROC curve - Area Under Curve(AUC) is also a widely used matrix for evaluation. It is used for binary classification problems. AUC of the classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example

F1 score is the harmonic mean between precision and recall. The range of F1 scores lies between [0,1]. It tells how precious your classifier is. (How many instances it classified correctly), as well as how robust it is

High precision but lower recall, gives an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 score, the better the performance of the model. Mathematically it can be expressed as,

$$F1\ Score = 2 * \frac{Precision * Recall}{Presision + Recall}$$

4.3 Evaluation plan

As described earlier evaluation of the model is conducted against the two data sets. To evaluate the performance of the features generated through feature engineering techniques we have used the initial transactional data set as the baseline. In the evaluation process, a 30% segment of the data set was considered for evaluating the performance of the predictive model. The results obtained are explained under section 4.4.1.

The other evaluation was done against the scoring index that we have generated. The evaluation of the scoring index for the generated features is described under section 4.4.2.

4.4 Evaluation of results

To evaluate the performance of the proposed model, we have evaluated the effect of feature generation and the performance of the scoring index.

4.4.1 Evaluating the performance of generated features

(Table 15) shows the comparison of the results obtained by the classifiers against the different feature sets. First, we have evaluated the performance of the classification matrices without applying any feature engineering technique and then have applied the feature engineering techniques and have observed the results.

(Table 14) shows the results obtained for the confusion matrix for the Random Forest classifier and XgBoost classifier for the initial transaction data set and for the data set after applying feature engineering techniques.

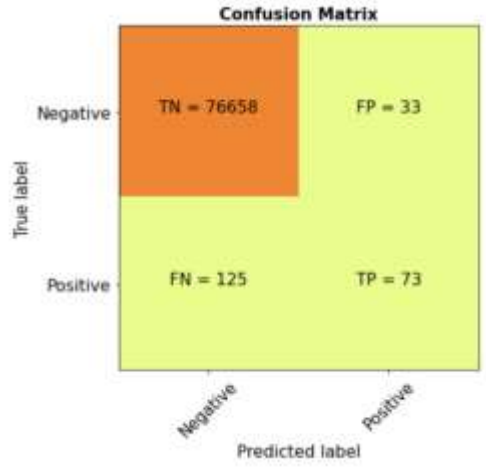
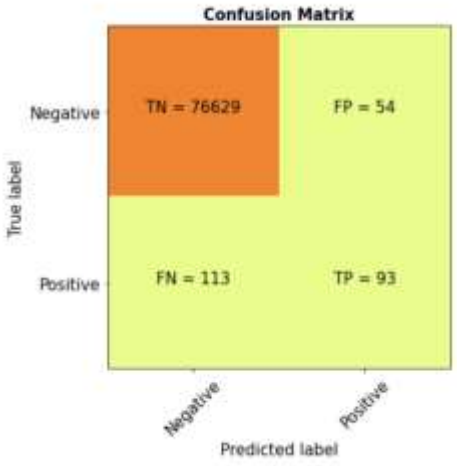
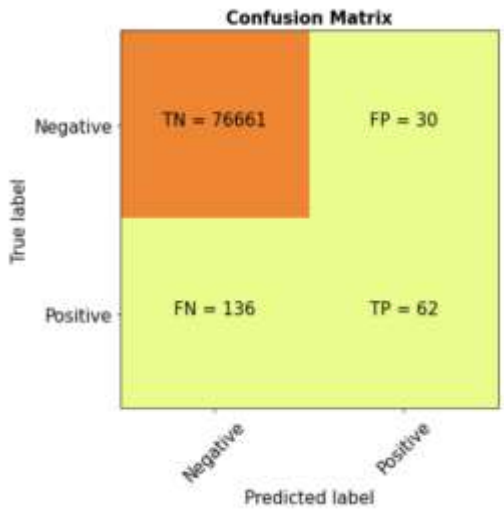
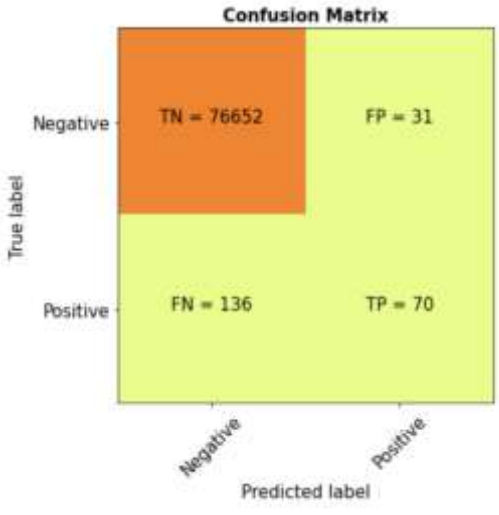
Confusion Metrix – Random Forest classifier	
Transaction Features	Feature engineering techniques
	
Confusion Metrix – XgBoost classifier	
Transaction Features	Feature engineering techniques
	

Table 14:Confustion matrix comparison

Since our data set is highly imbalanced the number of fraudulent transactions is (< 1%) compared to the legitimate transactions. We mainly consider the F1-Score and the AUC-ROC analysis. We can see that both of the classifiers have shown better performances after applying the feature engineering techniques. RF has shown a 10% increment in the F-Score while the XgBosst classifier has shown nearly 6% improvement in the F-Score. When it considers the ACU score both of the classification results have improved by 1%.

5. Classifiers	Transactional Features				
	F-measure	Precision	Recall	Accuracy	AUC
RF	0.49	0.71	0.37	0.99	0.95
XgBoost	0.43	0.67	0.31	0.99	0.98
Classifiers	Features generated through feature engineering techniques				
	F-measure	Precision	Recall	Accuracy	AUC
RF	0.54	0.64	0.47	0.99	0.96
XgBoost	0.46	0.69	0.34	0.99	0.99

Table 15: Performance of classifiers using different feature sets

(Figure 6) shows the performance of the AUC curve in the presence of different feature sets. There we can see that both classifiers have improved their performances after applying the feature engineering techniques and the XgBoost classifier has shown the best performances under a minimum false positive rate of 1% and obtained an AUC value of 0.996. Since the main intention of this research is to minimize the false-positive declines we have chosen XgBoost as the classifier for the proposed model.

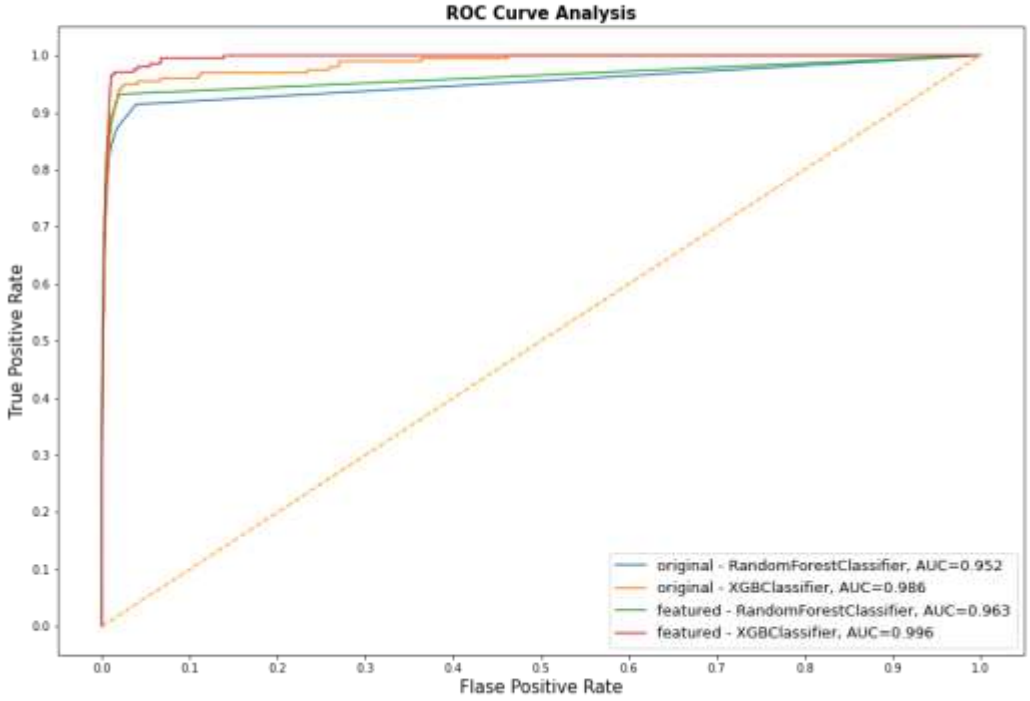


Figure 6:ROC curves obtained for different feature sets

4.4.2 Evaluating the scoring index

Our main research question is how to minimize false positives in CNP electronic payment systems. To answer our main research question we have come across two sub-research questions the second sub-research question is how to identify the best customers related to e-commerce Card Not Present (CNP) payment platforms? To answer this second sub-research question we came across the rule-based approach to generate a scoring index to identify the firms' best clients. Since there's no standard procedure to evaluate the performance we have manually validated the scores obtained by considering randomly chosen customers from each category. For. For Eg If a person has obtained a lower score based on the scoring index, we observed their weekly, monthly, and yearly transaction frequencies, the banker responses and the score obtained for the rule-based approaches, and the corresponding suspicious scores obtained by them based on the previous transaction history. Through this method, we have identified that frequent customers who have a clear transaction history have obtained a lower score, and also those who have been labeled as fraudulent transactions have obtained a high score through the rule-based approaches. In the other aspect, we have considered the

distribution of the scoring index given in Figure 10. We can accept the distribution where there exist only a few customers who have obtained a score <10 and most of the customers have obtained a score (20<score<30) because most of the customers have a transaction history of few transactions. And the distribution matches the generally acceptable distribution.

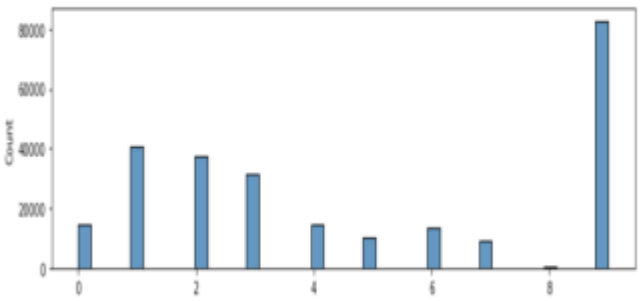


Figure 7 : Distribution of yearly critical index

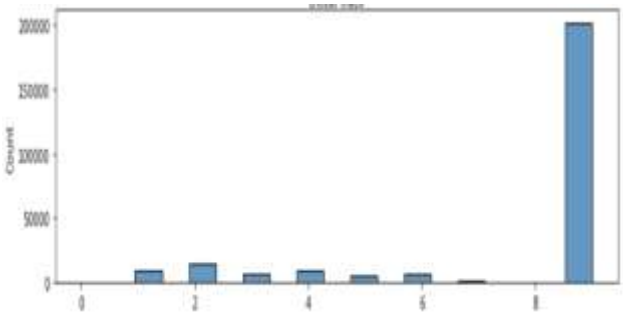


Figure 8:Distribution of monthly critical index

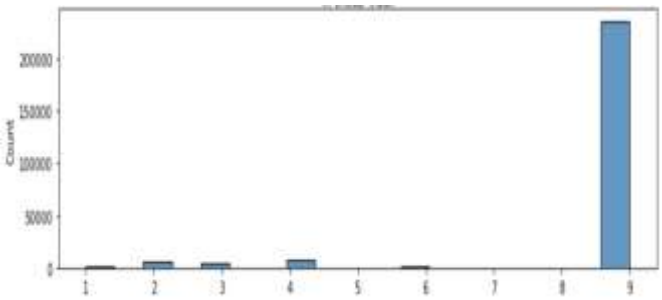


Figure 9:Distribution of weekly critical index

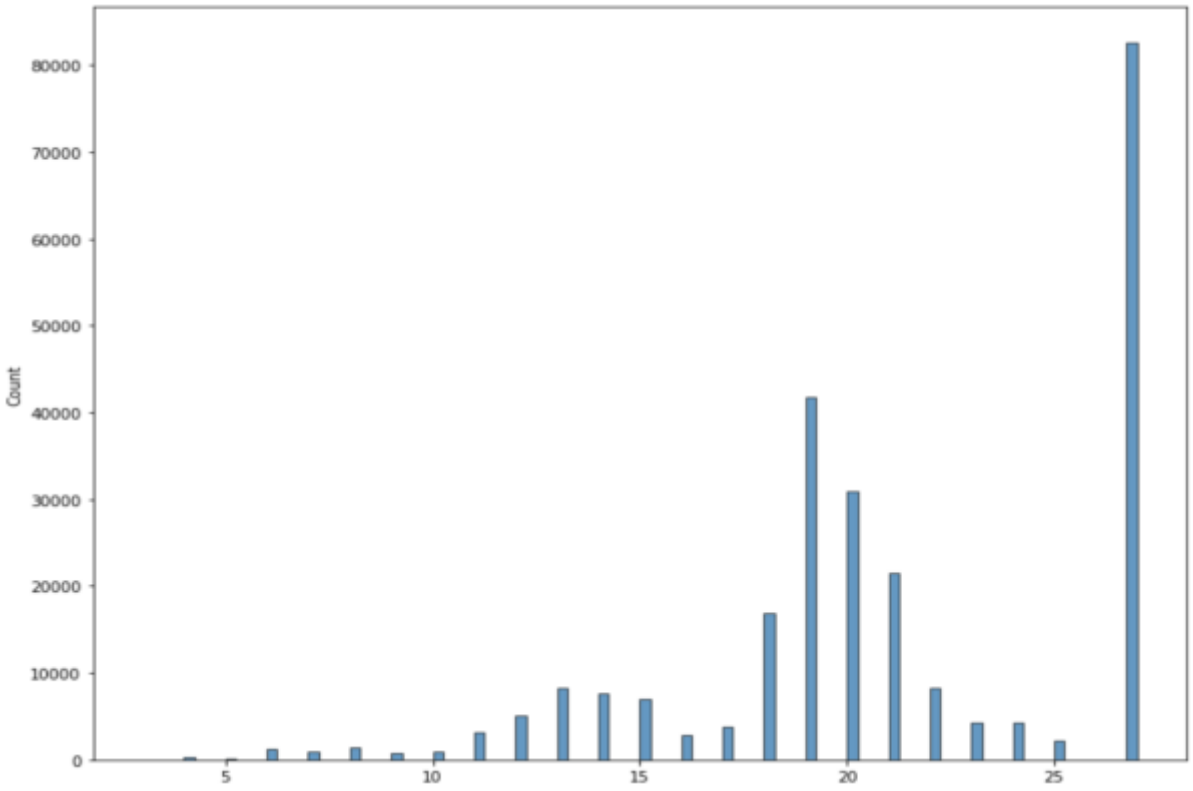


Figure 10:Distribution of final critical index

Based on the distribution identified three clusters of user category.

Scoring index	**User category**
Critical_index<10	Low risk
10<Critical_index≤20	Moderate
Critical_index > 20	High risk

Table 16: Defining user category

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In summary, in this paper, we propose a model to minimize the false-positive declines in electronic payment systems using feature engineering techniques.

Through the evaluation results, we can see that our proposed feature engineering technique provides better variables for the proposed model, and also the performances of the machine learning techniques can be improved by applying feature engineering techniques. We employed supervised learning techniques that have shown better performances in previous studies alongside behavior analysis. And we have found out that the XgBoost classifier has outperformed the Random Forest classifier related to the results obtained for the Auc score. Other than that the most important finding is that the scoring index that we have proposed can be used to identify the best customers in CNP electronic payment systems. Compared to the previous studies we have mainly focused on the heterogeneous behavior of CNP electronic payments and carried out behavior analysis.

This study has been conducted based on real-life data set from one of the fintech payment solution providers in Sri Lanka. The experimental results have shown that this proposed model can be used to minimize the false-positive declines as well as it can be used as a fraud detection system. In addition, we have provided a practical solution to overcome the false positive declines which becoming a severe problem for the service providers and the marginal analysis is that this proposed model can be used by payment solution providers to effectively and efficiently identify fraudulent transactions and minimize the fraud losses and protect the customer interests by providing a value-added service.

5.2 Limitations

But still, there are some limitations in this proposed feature engineering technique. Some of them are if the number of features that we initially select increases the generated features also increase exponentially. For Eg. Here we have considered only three aggregation periods as the weekly, monthly, and yearly if we add another aggregation period such as the last hour, last day then the features have to be generated separately for those periods as well. Providing too many features also reduces the performance of the classifier.

We have set some threshold values like small transaction amount, large transaction amount, time difference, etc. Currently, we have chosen those values based on the central bank transactional reports and related to the system transaction volumes. In future works, we plan to introduce a more convenient and generalized approach to automate the process of setting threshold values.

In this research to address the class imbalance problem and provide more meaningful values to the classifier, we have followed oversampling the minority by synthetically generated fraudulent transactions. We have synthetically generated instead of using SMOTE because SMOTE increases the minority by using the existing values and we have found out that it doesn't improve the classifier performances and cause problems like model overfitting. In future works, we plan to suggest a more convenient oversampling technique using GAN(Generative Adversarial Networks) to overcome the class imbalance problem.

5.3 Future Work

In our research, we have mainly selected a category of CNP electronic payment systems which is IPG transactions and evaluated the results and we intend to extend this method for other types like mobile wallet transactions and mPos transactions as well which is fallen into the CNP electronic payment category.

The other aspect is that we have applied the Supervised learning techniques to evaluate the performance of the model and we haven’t considered the behavior of the deep learning techniques in the presence of feature engineering techniques.

.

REFERENCES

A Major Challenge - False Positives, 2020. . FCASE. URL https://fcase.io/a-major-challenge-false-positives/ (accessed 1.10.21).

Chen, J., Shen, Y., Ali, R., 2018. Credit Card Fraud Detection Using Sparse Autoencoder and Generative Adversarial Network, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Presented at the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 1054–1059. https://doi.org/10.1109/IEMCON.2018.8614815

Consumer Sentinel Network, n.d.

Delecourt, S., Guo, L., 2019. Building a Robust Mobile Payment Fraud Detection System with Adversarial Examples, in: 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE). Presented at the 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 103–106. https://doi.org/10.1109/AIKE.2019.00026

Vesta, 2021. How to Avoid False Positives With Credit Card Fraud Detection [WWW Document], n.d. URL https://www.vesta.io/blog/false-positives-credit-card-fraud-detection (accessed 8.28.21).

Khatri, S., Arora, A., Agrawal, A.P., 2020. Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison, in: 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence). Presented at the 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence), pp. 680–683. https://doi.org/10.1109/Confluence47617.2020.9057851

Kim, E., Lee, J., Shin, H., Yang, H., Cho, S., Nam, S., Song, Y., Yoon, J., Kim, J., 2019. Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. Expert Syst. Appl. 128, 214–224. https://doi.org/10.1016/j.eswa.2019.03.042

Fraud.com. 2021. Kount_eBook_Silent_Sales_Killer_False_Positives.pdf, n.d.

Kumarage, T., Ranathunga, S., Kuruppu, C., Silva, N.D., Ranawaka, M., 2019. Generative Adversarial Networks (GAN) based Anomaly Detection in Industrial Software Systems, in: 2019 Moratuwa Engineering Research Conference (MERCon). Presented at the 2019 Moratuwa Engineering Research Conference (MERCon), pp. 43–48. https://doi.org/10.1109/MERCon.2019.8818750

Lei, S., Xu, K., Huang, Y., Sha, X., 2020. An Xgboost based system for financial fraud detection. E3S Web Conf. 214, 02042. https://doi.org/10.1051/e3sconf/202021402042

Lucas, Y., Portier, P.-E., Laporte, L., He-Guelton, L., Caelen, O., Granitzer, M., Calabretto, S., 2020. Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs. Future Gener. Comput. Syst. 102, 393–402. https://doi.org/10.1016/j.future.2019.08.029

Najadat, H., Altiti, O., Aqouleh, A.A., Younes, M., 2020. Credit Card Fraud Detection Based on Machine and Deep Learning, in: 2020 11th International Conference on Information and Communication Systems (ICICS). Presented at the 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 204–208. https://doi.org/10.1109/ICICS49469.2020.239524

Nami, S., Shajari, M., 2018. Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. Expert Syst. Appl. 110, 381–392. https://doi.org/10.1016/j.eswa.2018.06.011

Priscilla, C.V., Prabha, D.P., 2020. Influence of Optimizing XGBoost to handle Class Imbalance in Credit Card Fraud Detection, in: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT). Presented at the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1309–1315. https://doi.org/10.1109/ICSSIT48917.2020.9214206

Ryman-Tubb, N.F., Krause, P., Garn, W., 2018. How Artificial Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. Eng. Appl. Artif. Intell. 76, 130–157. https://doi.org/10.1016/j.engappai.2018.07.008

Singh, A., Jain, A., 2019. Adaptive Credit Card Fraud Detection Techniques Based on Feature Selection Method, in: Bhatia, S.K., Tiwari, S., Mishra, K.K., Trivedi, M.C. (Eds.), Advances in Computer Communication and Computational Sciences, Advances in Intelligent Systems and Computing. Springer, Singapore, pp. 167–178. https://doi.org/10.1007/978-981-13-6861-5_15

Grandi, M. 2021. The True Cost of False Positives in the Battle to Prevent Fraud [WWW Document], n.d. URL https://www.incognia.com/blog/the-true-cost-of-false-positives-in-the-battle-to-prevent-fraud (accessed 6.5.21).

Van Vlasselaer, V., Bravo, C., Caelen, O., Eliassi-Rad, T., Akoglu, L., Snoeck, M., Baesens, B., 2015. APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions. Decis. Support Syst. 75, 38–48. https://doi.org/10.1016/j.dss.2015.04.013

Wedge, R., Kanter, J.M., Veeramachaneni, K., Rubio, S.M., Perez, S.I., 2019. Solving the False Positives Problem in Fraud Prediction Using Automated Feature Engineering, in: Brefeld, U., Curry, E., Daly, E., MacNamee, B., Marascu, A., Pinelli, F., Berlingerio, M., Hurley, N. (Eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 372–388. https://doi.org/10.1007/978-3-030-10997-4_23

Xie, Y., Liu, G., Cao, R., Li, Z., Yan, C., Jiang, C., 2019. A Feature Extraction Method for Credit Card Fraud Detection, in: 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS). Presented at the 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), pp. 70–75. https://doi.org/10.1109/ICoIAS.2019.00019

Zhang, X., Han, Y., Xu, W., Wang, Q., 2019. HOBA: A Novel Feature Engineering Methodology for Credit Card Fraud Detection with a Deep Learning Architecture. Inf. Sci. https://doi.org/10.1016/j.ins.2019.05.023

Zhang, Y., Tong, J., Wang, Z., Gao, F., 2020. Customer Transaction Fraud Detection Using Xgboost Model, in: 2020 International Conference on Computer Engineering and Application (ICCEA). Presented at the 2020 International Conference on Computer Engineering and Application (ICCEA), pp. 554–558. https://doi.org/10.1109/ICCEA50009.2020.00122

Zhou, H., Chai, H., Qiu, M., 2018. Fraud detection within bankcard enrollment on mobile device based payment using machine learning. Front. Inf. Technol. Electron. Eng. 19, 1537–1545. https://doi.org/10.1631/FITEE.1800580

APPENDICES

APPENDIX A

Collecting the data set and apply basic filtering

```
class MultiColumnLabelEncoder:
    def __init__(self,columns = None):
        self.columns = columns # array of column names to encode

    def fit(self,X,y=None):
        return self # not relevant here

    def transform(self,X):
            output = X.copy()
        if self.columns is not None:
            for col in self.columns:
                output[col] = LabelEncoder().fit_transform(output[col])
        else:
            for colname,col in output.iteritems():
                output[colname] = LabelEncoder().fit_transform(col)
        return output

    def fit_transform(self,X,y=None):
        return self.fit(X,y).transform(X)

df = MultiColumnLabelEncoder(columns['issuer_bank', 'ip_address',
'browser']).fit_transform(df)


df['createdAt']=pd.to_datetime(df.createdAt)
df['createdAt'] = df.createdAt.values.astype(np.int64)
df['transaction_date']=pd.to_datetime(df.transaction_date)
df['transaction_time']=pd.to_datetime(df.transaction_time)
df['transaction_date'] = df['transaction_date'].apply(lambda x: x.value)
df['transaction_time'] = df['transaction_time'].apply(lambda x: x.value)
con_vars = ['payerAccountNumber','payerIdTransformed', 'payeeIdTransformed', 'createdAt',
'transaction_date' ,'transaction_time', 'issuer_bank', 'ip_address','browser']
df.fillna(0)

//Standardize the data set

scaler = StandardScaler()
df[con_vars]=scaler.fit_transform(df[con_vars])
df[con_vars]  =round(df[con_vars],10)
//save the data to a csv
df.to_csv('standerdize.csv',index=False)
```

Generate features based on aggregation characteristics

```
class FeatureAggregatorFunctions:
    #period 1-yearly, 2-monthly 3-weekly
    def __init__(self,period = 1):
        self.period = period

    def generate_features(self,x):
        data_process = x.groupby(['payerIdTransformed','payerAccNo'], as_index=False).agg({
        'createdAt': lambda x: (snapshot_date - x.max()).days,
        'payerAccountNumber': 'count',
        'originalAmount':'sum',
        'payeeIdTransformed':'count',
        'funding_method': 'sum',
        'secure_3ds':'sum',
        'success_3ds':'sum',
        'browser':'count',
        'status':'sum',
        'type':'sum',
        'currency':'sum',
        'amount_high':'sum',
        'suspicious_time':'sum',
        'high_risk_credit_transaction':'sum',
        'bank_approved':'sum',
        'bank_err':'sum',
        'bank_invalid_card':'sum',
        'bank_insufficient_fund':'sum',
        'bank_do_not_honor':'sum',
        'bank_lost_card':'sum'})

        data_process.rename(columns={'createdAt': 'recency',
                    'payerAccountNumber': 'frequency',
                    'originalAmount': 'total_value',
                    'payeeIdTransformed':'merchant_frequency',
                    'funding_method': 'tot_credit_card_trans',
                    'secure_3ds':'secure_3ds_count',
                    'success_3ds':'success_3ds_count',
                    'browser':'diff_brouser_count',
                    'status':'success_count',
                    'type':'one_time_count',
                    'currency':'lkr_trans',
                    'suspicious_time':'tot_suspiciout_tm_trans',
                    'high_risk_credit_transaction':'tot_high_risk_credit_trans',
                    'bank_approved': 'tot_bank_approves',
                    'bank_err': 'tot_bank_err',
                    'bank_invalid_card':'tot_invalid_card',
                    'bank_insufficient_fund':'tot_insufficient_fund',
                    'bank_do_not_honor':'tot_bank_do_not_honor',
                    'bank_lost_card':'tot_bank_lost_card',
                    'amount_high':'tot_amount_high'}, inplace=True)
        data_process['unique_id'] = data_process.payerIdTransformed.astype(str) + '_' +
data_process.payerAccNo.astype(str)
```

```
    return data_process
```

Application of rule-based approaches

```
transaction_count_threshold=10
class FeatureRules:
    #period 1-yearly, 2-monthly 3-weekly
    def __init__(self,period = 1):
        self.period = period


    def success_tans(self,x):
        names = {'tot_success_amt': x[x['status']
==1]['originalAmount'].sum().round(2)}
        return pd.Series(names, index=['tot_success_amt'])


    def suspicious_failure_rate(self,x):
         if ((100 - x['success_percent']) > 50):
          return 1
         return 0

    def recurring_suspicious_banker_response(self,x):

tot_error=x[['tot_bank_err','tot_invalid_card','tot_insufficient_fund','tot
_bank_do_not_honor','tot_bank_lost_card']].sum()
         if(tot_error>0 and x['frequency']> transaction_count_threshold ):
           if(((x['frequency']-x['one_time_count'])>0) and
((round((x['tot_insufficient_fund']/tot_error)*100,2) <50))):
             return 1
           if(((x['one_time_count'])>0) and ((100-
x['success_percent'])>50)):
             return 1
         return 0

    def check3ds(self,x):
        if((x['secure_3ds_count']-x['success_3ds_count'])>0 and
x['one_time_count'] > 0):
            return 1
        elif(x['secure_3ds_count']==0 and x['one_time_count'] >0):
            return 1
        return 0

    def suspicious_tm_transaction(self,x):
        if(x['tot_suspiciout_tm_trans']>1 and x['one_time_count']>0):
         return 1
        return 0

    def suspecious_credit_trnasaction(self,x):
        if(self.period==1 and x['frequency'] > 0):
            if((x['tot_high_risk_credit_trans']/x['frequency'])*100 > 50):
             return 1
        elif(x['tot_high_risk_credit_trans']>1):
         return 1
        return 0
    def suspecious_card_usage(self,x):
        if(x['same_card_diff_user_count']>3):
         return 1
```

```
        return 0
    def calculate_time_diff(self,x, y):
        y = datetime.strptime(y, '%H:%M:%S')
        x = datetime.strptime(x, '%H:%M:%S')
        time_delta = (y - x)
        total_seconds = time_delta.total_seconds()
        minutes = total_seconds/60
        return minutes

    def covert_column(self,x, col):
        if 1 in x[col].tolist():
            return 1
        return 0

    def apply_time_diff(self,x, method = "default"):
        time_threshold = 15
        high_threshold = 10000
        test_threshold = 1000

        amounts = x['originalAmount'].tolist()
        higharray = x['amount_high'].tolist()
        timearray = x['transaction_time'].tolist()

        if len(timearray) > 1:
            lastitem = None
            index = 0
            total = 0
            for i in timearray:
                if lastitem != None:
                    if method == "high_amounts":
                        close_transactions =
self.calculate_time_diff(lastitem, i) <= time_threshold
                        if close_transactions and higharray[index]: return
1
                    elif method == "high_sum":
                        time_threshold = 60
                        close_transactions =
self.calculate_time_diff(lastitem, i) <= time_threshold
                        total = total + amounts[index]
                        if close_transactions and total >= high_threshold :
return 1
                    elif method == "test_transactions":
                        time_threshold = 120
                        close_transactions =
self.calculate_time_diff(lastitem, i) <= time_threshold
                        if close_transactions and amounts[index-1] <=
test_threshold and amounts[index] >= test_threshold: return 1
                    else:
                        close_transactions =
self.calculate_time_diff(lastitem, i) <= time_threshold
                        if close_transactions: return 1
                lastitem = i
                index = index + 1

            return 0
        else:
            return 0
```

Evaluation of the model performances

```
result_table = pd.DataFrame(columns=['classifiers', 'fpr','tpr','auc'])
for name,model in models:
    kfold = KFold(n_splits=10, random_state=1,shuffle=True)
    cv_result = cross_val_score(model, X_train, y_train, cv = kfold,
scoring = "accuracy")
    roc_auc_result = cross_val_score(model, X_train, y_train, cv = kfold,
scoring = "roc_auc")
    print(model.__class__.__name__,)
    print('\n')
    print(name, cv_result)
    print("=== Mean Accuracy ===")
    print("Mean Accuracy Score -: ", cv_result.mean())
    print('\n')
    print(name, roc_auc_result)
    print("=== Mean AUC Score ===")
    print("Mean AUC Score -: ", roc_auc_result.mean())
    print('\n')
    model.fit(X_train, y_train)
    # predictions
    actual=y_test
    predict = model.predict(X_test)

    print("=== Confusion Matrix ===")
    cm = confusion_matrix(actual,predict)
    print(cm)
    print('\n')

    plt.figure(figsize=(8,6))
    plt.clf()
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Wistia)
    classNames = ['Negative','Positive']
    plt.title('Confusion Matrix',fontsize=15,fontweight='bold')
    plt.ylabel('True label',fontsize=15)
    plt.xlabel('Predicted label',fontsize=15)
    tick_marks = np.arange(len(classNames))
    plt.xticks(tick_marks, classNames, rotation=45,fontsize=15)
    plt.yticks(tick_marks, classNames,fontsize=15)
    s = [['TN','FP'], ['FN', 'TP']]
    for i in range(2):
        for j in range(2):
            plt.text(j,i, str(s[i][j])+" =
"+str(cm[i][j]),horizontalalignment="center",fontsize=15)
    plt.show()

    print("=== Classification Report ===")
    print(classification_report(actual,predict))
    print('\n')

    importances=model.feature_importances_
    fi = pd.DataFrame({'feature': list(X_train.columns),
                   'importance':
model.feature_importances_}).sort_values('importance', ascending = False)

    # Display
    print(fi)


    probs = model.predict_proba(X_test)[:, 1]
    fpr, tpr, thresholds = roc_curve(actual,  probs)
    roc_auc = roc_auc_score(actual,  probs)
```

```
    result_table = result_table.append({'classifiers':"original -
"+model.__class__.__name__,
                                        'fpr':fpr,
                                        'tpr':tpr,
                                        'auc':roc_auc}, ignore_index=True)
result_table.set_index('classifiers', inplace=True)
```