National College of Ireland

Comparative Analysis of Batch and Online Machine Learning Techniques for Fraud Detection: A Case Study on Real European Credit Card Transactions

MSc Research Project
Data Analytics

Saran Raj Srinivasan
Student ID: x22149066

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Saran Raj Srinivasan
Student ID:	x22149066
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Noel Cosgrave
Submission Due Date:	31/01/2024
Project Title:	Comparative Analysis of Batch and Online Machine Learning Techniques for Fraud Detection: A Case Study on Real European Credit Card Transactions
Word Count:	8190
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Saran Raj Srinivasan
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Batch and Online Machine Learning Techniques for Fraud Detection: A Case Study on Real European Credit Card Transactions

Saran Raj Srinivasan
x22149066

Abstract

Machine learning has always played a important role in fraud detection, with different methods developing to meet the constantly changing strategies of the fraudsters. In the context of "credit card fraud detection" this study, addresses the difficulties caused by missing data and class imbalance as well as comparing and analysing the performance of batch and online machine learning algorithms. The study utilizes the data set from Vesta Corporation managed by the IEEE Computational Intelligence Society from Europe, which contains real credit card transactions and is marked by a high level of class imbalance and 41percentage missing values. These missing values are handled using multiple imputation, and the imbalance is addressed using a hybrid method that combines SMOTE and random under sampling. The results demonstrate that online machine learning techniques, particularly Online Random Forest and Online XGBoost, outperform their batch methods. The study concludes that online machine learning models are more effective in handling imbalanced data sets with missing values and recommends further research to validate these findings on other data sets and domains.

1 Introduction

The increasing likelihood of financial fraud is one of the challenging issues brought about by the development of digital banking services and the rising use of online card transfers. This research papers focus on the importance of machine learning methods in financial fraud detection Sadgali et al. (2019).The trust that customers have in banking services can be reduced by fraudulent activity, which can lead to large financial losses for financial institutions as well as customers. To keep up with the dynamic nature of fraud trends, the fraudsters ever-changing strategies, and the imbalanced structure of transaction data Gupta et al. (2023) in real data set it became very difficult in handling and improving the fraud detection algorithms. Hence in this research address the topic of comparing batch machine learning against equivalent, online learning techniques in the context of credit card fraud detection with real data set. Burlutskiy et al. (2016) This research intends to investigate the possibilities of deep learning techniques and machine learning models along with providing the important insights on handling class imbalance and missing value in real data.

1.1 Research Question

"How can a comparison between batch and online machine learning techniques can reveal their strengths and weaknesses in the context of fraud detection in real credit card transactions, while effectively addressing the challenges posed by missing values and class imbalance in the data set"

1.2 Research Objectives

The objective of the research project is to analyse and study how different machine learning model performs on the real data set, hence the below experiments and data set selections are made:

The IEEE-CIS Credit Card Fraud Detection data set, which is accessible on Kaggle, was selected for the research project. Real transactions involving credit cards performed by European cardholders in September 2013, are included in the data set providing a significant challenge in terms of developing an efficient fraud detection system due to its high imbalance and 41percentage of missing values.

Creating an approach by reviewing various research work on online machine learning, addressing imbalance, and missing valuesHasan et al. (2021).

Comparison between the models built from Batch and Online Machine Learning.

This research contributes the clear-cut comparison between batch and online learning when applied to predicting fraud detection with real credit card transaction data set.

1.3 Structure of the Report

The opening Section 1 highlights the research challenge, examines the subject issue, and highlights the significance of the topic. A comprehensive analysis of previous research is provided by the literature review in Section 2. Section 3 discuss methodology, outlines the processes for data collection and preparation, chosen machine learning models for fraud detection. Section 4 covers Design specification. Section 5 shows the Implementation, Section 6 shows Evaluation and Results. The Section 7 Discussion, assesses the results. The Last Section 8 outline the Conclusion and Future works.

2 Literature Review

The choice of between batch and online machine learning methods is essential when using machine learning, which is a common way for detecting fraud. Batch learning, often called "offline learning," requires instantaneous access to all data. However, because of memory limitations, it is not appropriate for large-scale data sets and scenarios with evolving patterns over time. Online learning, on the other hand, also known as incremental or data stream learning, works with sequential data and time limitations since it analyzes and learns from data record by record.

2.1 Introduction to Batch and Online Machine Learning

Batch learning, in which the system is trained using all of the available data set offline. It doesn't gradually adjust to newly received data. The steps involved in batch learning

include deployment, training, and data gathering. Bisong and Bisong (2019) It might be difficult to do batch learning on big data sets as it usually takes a lot of CPU, memory, and storage capacity.

On the other hand, online machine learning uses real-time sample data, one observation at a time. Hoi et al. (2021) With more useful batch algorithms, online learning models handle a single sample of data at a time, greatly increasing their efficiency in terms of both time and space. Online learning may be helpful in situations where samples are provided gradually and it is expected that the data' distribution of chances will change over time. It follows that the model must also be updated often to take such changes into account and make necessary adjustments.

2.1.1 An crucial analysis of Batch vs Online Machine Learning

An investigation, comparing online and batch (offline) learning algorithms for user behavior prediction is presented in Burlutskiy et al. (2016) and focuses on real-time predictions. The paper tackles the issues of accuracy and structural complexity in real-time predictions, by framing those problem of user behavior prediction as a classification task. Using data from the Stack Overflow platform, the study experimentally investigates the performance of online and batch algorithms and suggests a way for comparing them. These findings show that a simple online learning algorithm works just as well as a Neural networks, a deep learning algorithm, and surpasses modern batch techniques. This study sheds light on how well online learning works in real-time to predict user behavior and how well it performs in comparision to batch learning algorithms.

Batch learning and Online learning, are the two design options for how data is utilized in the modeling pipeline, and they are both covered in Bisong and Bisong (2019). In online learning, the inputs were entered from the learning algorithm in streams, either as individual sample points or in tiny batch sizes, whereas in batch learning, the machine learning model is trained using the complete data set that is accessible at a given moment. The article offers a brief overview of these ideas, stressing the distinctions between online and batch learning as well as the uses for both. It is an invaluable tool for understanding how these two learning techniques affect model deployment and training.

2.2 A Review on Fraud detection with Machine learning

A real-time credit card fraud detection system employing machine learning approaches is presented in Thennakoon et al. (2019). The use of machine learning techniques, such as XGBoost and Random Forest, to identify fraudulent transactions in real time is the main topic of this article. Incoming credit card transactions are processed by the system, which also looks for possible fraud details based on transaction quality and historical data. The study shows how well the suggested method works in real time at identifying fraud by evaluating its performance using actual credit card transaction data. The study emphasizes how important real-time fraud detection is for preventing financial losses and preserving credit card transaction security.

The implementation of gradient boosting techniques in credit card fraud detection is examined in Ramani et al. (2022). With an focus on the use of Cat Boost, and Light Gradient Boosting Machine (LGBM) for credit card fraud detection, the paper offers a thorough review of several supervised, and unsupervised machine learning algorithms for

identifying fraudulent actions. The use of these methods and how well they detect fraudulent transactions are covered in this study. These findings show that 'gradient boosting' approaches have the ability to increase the precision of credit card fraud detection, underscoring their importance in tackling the problems associated with fraudulent activity in financial transactions.

A new ensemble method, for predicting fraud in credit card transactions is presented in Baabdullah et al. (2022). The suggested technique known as Man-Ensemble CCFD, makes use of an ensemble-learning model that goes through two phases of feature selection and classification. The most relevant qualities are chosen in the first step using a random forest classifier, and fraud is predicted in the second stage using a gradient boosting classifier. The suggested method's efficiency is analyzed in the study using two real data sets of credit card transactions, both Non-fraudulent and fraudulent. The findings show that the suggested strategy beats a number of fraud detection methods, such as random forest, logistic regression, and decision trees. The study emphasizes how ensemble-based machine learning may increase the precision of credit card transaction fraud detection.

2.3 A critical review on handling Real Data set for fraud detection

2.3.1 Handling Missing Value

The issues in missing data and the methods for addressing them are covered in Kang (2013) and Hasan et al. (2021). The researcher highlights the many approaches to handling missing data, such as imputation and multiple imputation, and offers a thorough overview along with the analysis of the works done on Missing Value Imputation (MVI) from 2010 to 2021. This highlights how crucial it is to collect and organize study-related data accurately in order to prevent missing data, and also talks about the imputation technique which includes replacing missing data with approximated values. These studies offer insightful explanations of the difficulties associated with missing value imputation and how they affect the effectiveness of machine learning models.

The effect of feature selection on missing value imputation in medical data sets is analysed in Liu et al. (2020). The main focus of this study is wrapper techniques, which depend on a parameter (or learning model) as their aim for assessing different feature subsets. The impact of feature selection on medical data set imputation of missing values is examined in this work, with a focus on machine learning and predictive modeling. The quality and precision of the imputed values are greatly impacted by the feature selection, which is why the authors stress its significance in the imputation process.

Heymans and Twisk (2022) offers insights on the difficulties associated with missing data in clinical research as well as methods for addressing it. The research highlights how important it is to fill in the gaps in clinical trial data as they might lead to skewed calculations and compromise the reliability and inter pretability of test outcomes. The authors offer programming approaches and strategies to deal with missing data, emphasizing multiple imputation in particular. Applying a set of actual values in place of missing data, multiple imputation accounts for error and variability that occurs. This preserves all instances and increases the analysis accuracy.

2.3.2 Handling Imbalanced data set

An innovative hybrid strategy to handle unbalanced data sets is presented in Desuky and Hussain (2021), which was published in the Arabian Journal for Science and Engineering in 2021. The suggested technique employs support vector machines and under samples using the simulated annealing methodology. The study, which involved 11 data sets, provides competitive results. The suggested approach demonstrated a balanced performance on a range of unbalanced data sets. The paper addresses the important problem of class imbalance, especially in real data sets, and offers a workable and efficient way to lessen the negative effects of imbalance on machine learning models. The method demonstrates how important it is to combine under sampling and oversampling strategies in order to handle uneven data in a balanced and accurate manner.

Chawla et al. (2002) presents a solution to the machine learning issue of class imbalance. The imbalance that results when real-world data sets contain a large proportion of normal cases and relatively few abnormal or minority ones is the main topic of the study. The suggested method entails synthesizing samples of the minority class and oversampling it, In order to improve classifier performance. To increase the minority class, the approach creates synthetic examples that are similar to instances of the minority class that already exist. The study shows that this strategy can lessen the negative effects of class imbalance on machine learning models and enhance classifier performance, especially in ROC(Receiver Operating Characteristic) space.

2.4 Research Gaps and Conclusion of Literature

The literature analysis highlights that, given the rise in fraud cases and the constantly changing pattern of fraudulent operations, fraud detection in digital transactions is vital. The efficiency of fraud detection algorithms is greatly impacted by the occurrence of large missing values and higher class imbalance in real bank transaction data sets. There is a evident lack of research explicitly addressing fraud detection in banking transactions using the real transaction data set with high missing values, despite the fact that previous studies have shown the potential of machine learning techniques, such as gradient boosting and ensemble learning, in credit card fraud detection.

Further studies may be done to investigate how well machine learning approaches handle missing values and class imbalance in data sets of actual bank transactions, in order to fill in the research gaps further. For example, banking transactions can be covered by Thennakoon et al. (2019) study on real-time credit card fraud detection using machine learning. In a similar way, real bank transaction data sets may be used to investigate the efficiency of batch vs online learning methods in Bisong and Bisong (2019) study on the subject of immediate fraud detection.

In summary, these current work intends in comparing online and batch machine learning along with different techniques using a real transaction data set and approaches to solve the problems caused by the data set missing values and class imbalance. The study intends to offer important insights into the creation of more reliable and efficient fraud detection systems by utilizing a real bank transaction data set.

3 Methodology

The purposeful utilization of data analysis and/or logical approaches to define, compress, and analyze data is known as data analysis, and it is an essential procedure. There are several potential methodology frameworks to take into account. These include the Knowledge Discovery in Databases (KDD) paradigm and the Cross Industry Standard Process for Data Mining (CRISP-DM) in particular. This research work on handling missing values and imbalanced data sets in a real banking data set which aligns well with the KDD methodology Fig. 1. The research work can be summarized as follows:

1) Choosing a real fraud transaction data set;

2) Understanding transaction data;

3) Preparation transaction data;

4) Modeling, and Evaluation, which includes Comparing batch and online machine learning to learn and get insights.

This emphasizes how methodology and design interact.

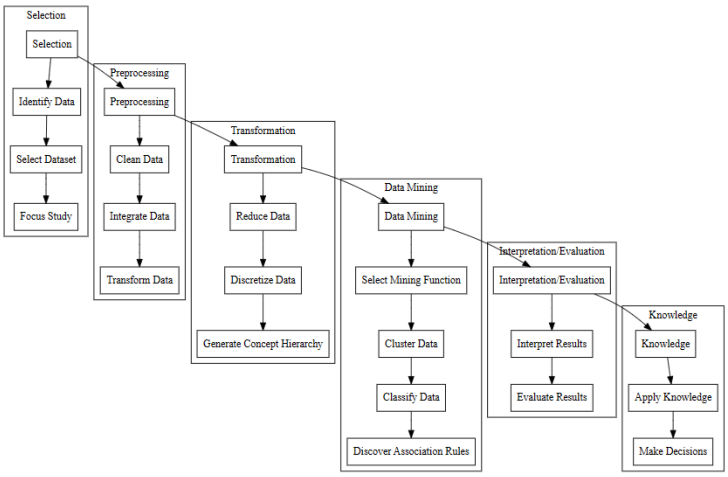

Figure 1: KDD flow chart

3.1 Selection of Data set

The IEEE-CIS Fraud Detection data set, Fig. 2 which is accessible on Kaggle [1], was selected for the research project.The data set is provided by Vesta Corporation and is prepared by IEEE Computational Intelligence Society. This data set is primarily designed to support in the creation of models and algorithms for fraud detection, which will help detect possibly fraudulent transactions. Transactions involving credit cards performed by European cardholders in September 2013 are included in the data set. The transactions in this data set of card-not-present transaction fraud are completed without the cardholder's physical presence at the point of sale. An information shows transactions that took place over a two-day period, with 492 out of 284,807 transactions being fraudulent. The positive class (frauds) consists nearly 0.172 percentage of all transactions in the data set, which is extremely unbalanced. The data set consists of 434 columns, 431 anonymized

[1]https://www.kaggle.com/competitions/ieee-fraud-detection

features, ranging transaction ID, transaction amount, and transaction time. There are 41 percentage missing values in the data set, which consists of 590,540 rows.

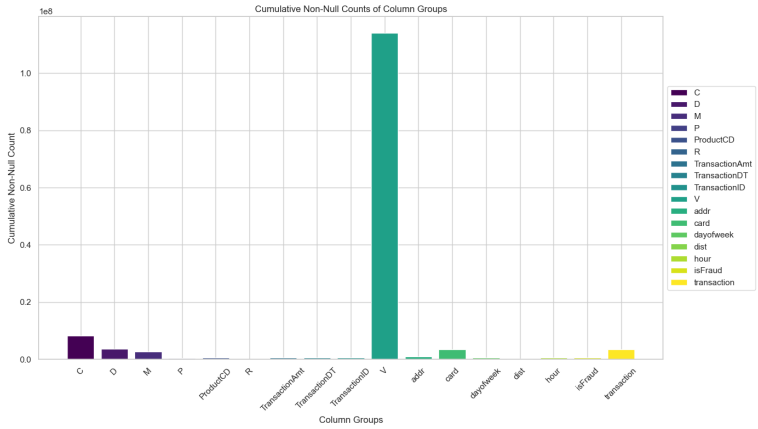

Figure 2: IEEE Dataset

3.1.1 Data set Description:

Column	Description
TransactionDT	Time delta from a given reference date time
TransactionAMT	Transaction payment amount in USD
ProductCD	Product code, the product for each transaction
card1 - card6	Payment card information, such as card type, card category, issue bank, country, etc.
addr	Address of the purchaser
addr1	Billing region
addr2	Billing country
dist	Distance: distances between billing address, mailing address, zip code, IP address, phone area, etc.
P and (R) Email domain	Email domain, purchaser and recipient email domain
C1-C14	Counting, such as how many addresses are found to be associated with the payment card, etc.
D1-D15	Timedelta, such as days between previous transaction, etc.
M1-M9	Match, such as names on card and address, etc.
V1 - V339	features, including ranking, counting, and other entity relations. How many times the payment card associated with an IP and email or address appeared in a 24-hour time.

Table 1: Description of Transaction Dataset

3.1.2 EDA: Exploratory Data Analysis

Out of the 590,540 card transactions in the IEEE-CIS Fraud Detection data set, 20,663 (3.5percentage) are fraudulent. Every transaction includes a label indicating whether

it was fraudulent or valid, a relative timestamp, and 431 characteristics (400 numerical, 31 categories). The identities of the extra features that Vesta designed have been disguised, as have their names, in order to maintain privacy. It offers a vast amount of real-world transaction data that can be used to train and evaluate machine learning models. Relative timestamps and category and numerical variables are among the data sets characteristics. With just 3.5percentage of the transactions being fraudulent, the data set is wildly skewed. A section of the EDA visualization with important details to give attention to is shown below. The below shows the visualization of the TransactionDT and TransactionAmt column,

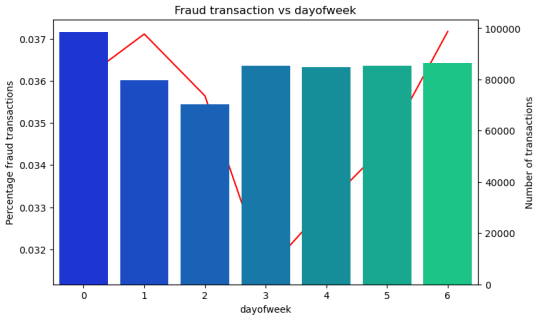

Figure 3: TransactionDT - Weekly

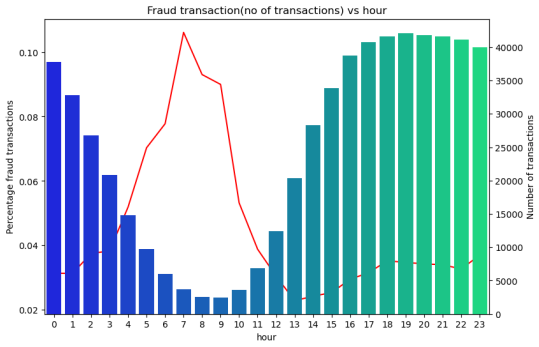

Figure 4: TransactionDT - Per hour

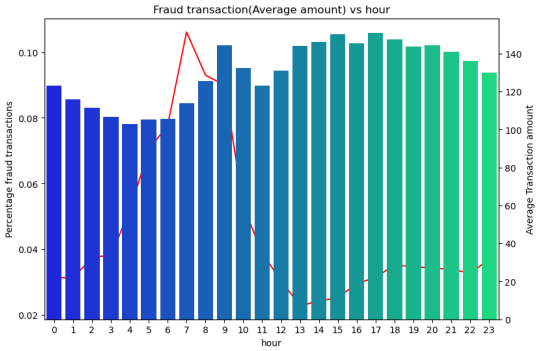

Figure 5: Transaction Amt

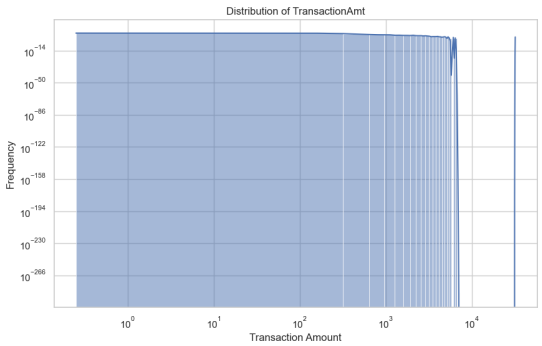

Figure 6: Transaction Amt distribution

TransactionDT is not a real timestamp; rather, it is a time delta from a specified reference date and time. As an example, 86400 can represent 86400 seconds from a certain time, such as from 12:05:09 pm on February 12, 2010. For security and privacy concerns, the precise time and date of the transactions are not disclosed. The TransactionDT function may be utilized to examine transaction timing patterns and spot any tendencies in fraudulent activity over time. Here in the Fig. 4 its examined how fraudulent transactions are distributed over time, like by day of the week or by hour of the day where the line graph of the percentage of fraudulent transactions and a bar graph of the number of transactions for each day of the week Fig. 3 and per hour. Further analysis can assist in identifying any certain times of day when fraud is most likely to happen.

Any unusual or unclear transaction amounts that can point to fraudulent activity can be found by analyzing the TransactionAmt feature's distribution Fig. 5. The transaction value in cash equivalent terms is given by this numerical feature. In the bar plot, a line graph shows the percentage of fraudulent transactions and a bar graph shows the average transaction amount for each hour of the day. According to the histogram Fig. 6,

the majority of transactions are small in size and rapidly decline in value as transaction amounts rise. A few very large transactions are often accompanied by a huge number of smaller ones in transaction data. New features, such as time-based connects or features that record the amount of time since the previous transaction, can be developed using the TransactionDT and TransactionAmt features.

'C' Column: It has a collection of 14 anonymized attributes that are connected to the card information of the transactions. It may be used to examine patterns of card-related information in transactions and spot any unusual or suspect patterns that can point to fraudulent activity. The linear correlation coefficients between characteristics 'C1' through 'C14' and the 'isFraud' variable are shown in the bar plot below Fig. 7. A direct association is shown by positive values, and an inverse relationship is indicated by negative values. The features with the highest positive correlations with 'isFraud' are 'C2', 'C8', and 'C12', whereas the features with the strongest negative correlations are 'C9' and 'C5'.

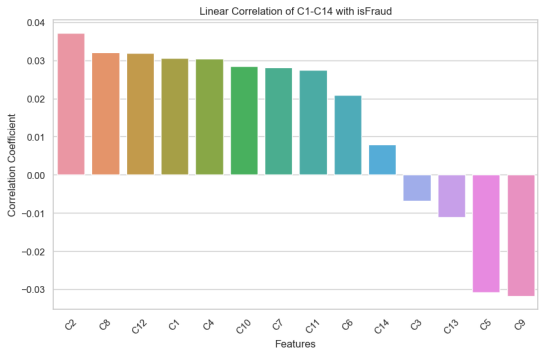

Figure 7: C column vs Fraud

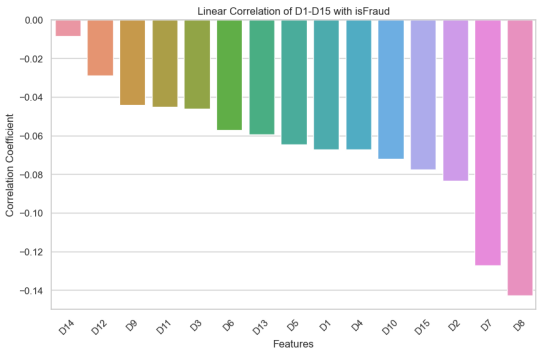

Figure 8: D column vs Fraud

'D' Column Fig. 8: Like the 'C' column 'D' is also related to the card information of the transactions. Each 'D' feature's value counts show the distribution of values; some negative values are present and may require additional study. A calculation and visualization of the linear correlation with 'isFraud' towards features D1–D15 from the above bar plot its visible that D7 and D8 had the highest negative correlation and some 'D' features have a small negative correlation with 'isFraud,' with D14 having the lowest negative correlation.

Product Code: Among different product categories in transactions the Product CD function can also be utilized to spot any trends or anomalies that can suggest fraud. The distribution of the 'Product CD' feature is as follows: Product Code W: 439,670 transactions (74.45percentage), Product Code C: 68,519 transactions (11.60percentage), Product Code R: 37,699 transactions (6.38percentage), Product Code H: 33,024 transactions (5.59percentage), Product Code S: 11,628 transactions (1.97percentage)

The bar plot Fig. 9 visualizes the number of transactions for each product code, showing that Product Code W is the most common by a significant margin. indicating that Product Code W is, by far among all transactions, is the most prevalent, as can be seen. There is also a higher count of product W in fraudulent transactions. However, fraud accounts for just 2 percent of all Product W transactions. Approximately 11.5 percent of all Product C purchases include fraud. Around six percent of Product S purchases are fraudulent as well.

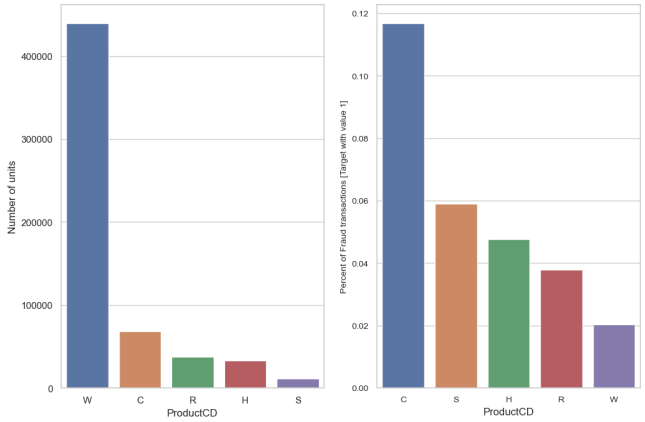

Figure 9: Details on Product Code

The card holder's address is represented by the addr1 and addr2 columns. Addr1 and Addr2 have 332 and 74 unique values, respectively, indicating that they are categorical features with several categories. The addr1 and addr2 columns contain a significant proportion of missing data, with 52.1percentage and 76.8percentage missing values, respectively, since they may not be accessible for a large number of transactions. Certain addr1 and addr2 value combinations could be more likely to be connected to fraud than others. Different aspects of location are represented by the addr1 and addr2 features; combining them to create another feature like a geographic feature might yield more information.

Card1-Card6: Anonymized data pertaining to the payment card used in the transaction is represented by the Card 1-6 columns in the data set. The numbers in the Card 1-6 columns range widely: the range for Card 1 is 1000 to 18396, the range for Card 2 is 100 to 600, and the range for Card 3 is 0 to 231.

P and R email domain: Email domains could be more frequently linked to fraudulent transactions, or certain combinations of email domains might be more likely to be connected to fraudulent activity. Protonmail.com had the greatest fraud rates, based to the exploratory data analysis of the P and R email domain columns and its relationship with isFraud. A very high fraud rate for both the purchaser and the recipient was found in this domains.

3.2 Pre-Processing of Transaction Data

3.2.1 Handling the missing value

It is crucial to evaluate the amount of missing values, the pattern of missing data, and the process underlying the missingness of the data in the context of this real data set, which includes 41percentage of missing values and is skewed toward non-fraudulent transactions. Wrapper subset selection is one method which deals with missing data by reducing the dimensionality; in this method, columns that share comparable missing values are merged, and only the columns that pose a prediction value are chosen. This method has decreased the data set's dimensionality and increased the analysis's effectiveness. In the Fig. 10 Heatmap the yellow region shows the missing value in the data set.

The missing pattern in the data set is analyzed to figure out the reason causing the

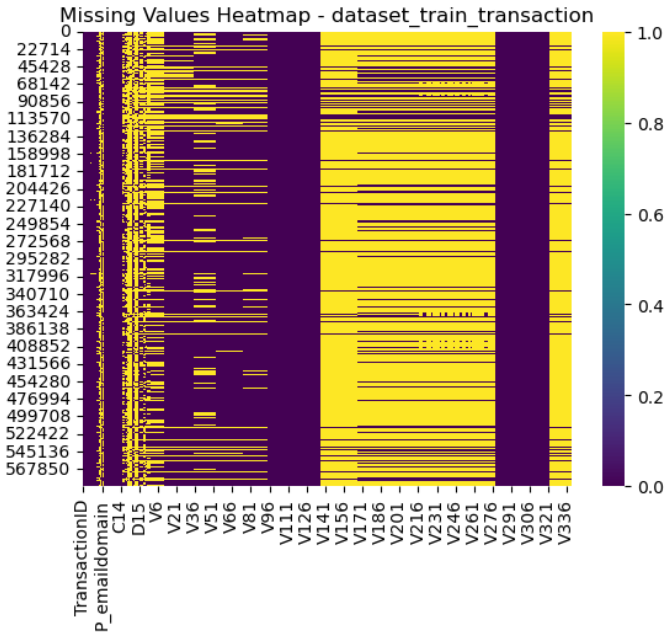

Figure 10: Heatmap of Missing value

missingness once the right columns have been chosen. Some imputation techniques are suitable depending on the missing data mechanism (MCAR, MAR, or MNAR)Lee et al. (2023). When missing data is MCAR, it indicates that there is no correlation between the missingness and any known or unknown variables its entirely random. If there is a consistent link between the missingness and other variables and the missing data is not MCAR. In the event that the missing data is MAR, the missingness is connected to the variables that have been observed rather than the missing data itself. In the event that the missing data is MNAR, the missingness is connected to the actual missing data.

In the below Fig. 11, 'Blank Spaces' stand in for missing values. Each column is a feature or variable, and each row represents a sample (in this example, 1000 randomly sampled rows). 'Bar to the Right': A line plot illustrating the completeness of each row is displayed in the bar to the right of the matrix.

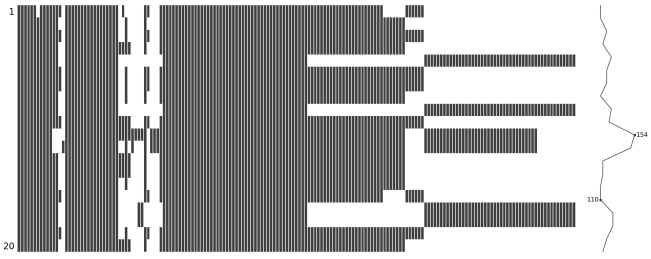

Figure 11: Missing Pattern MNAR

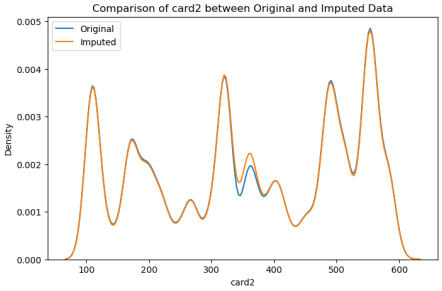

Figure 12: Multiple Imputation

It displays the number of values in each row that are not missing. In terms of available data, the row is larger the longer the bar. 'Patterns': Analyze any structures or patterns

in the gaps between rows or columns. If information consistently disappears from certain columns or rows, it may indicate a presence of a particular missing data mechanism. By analyzing these, there's a noticeable pattern between the missing values in different columns indicating Missing Not At Random (MNAR). Multiple imputation Fig. 12 approaches can be utilized to determine the missing values based on the observed data in order to address missing data for MNAR. The process of multiple imputation involves creating many real imputed data sets, each with an individual set of imputed values, and thereafter evaluating each data set independently. Post these handling process the actual data set column is reduced to 177 from 394.

3.2.2 Handling the Class Imbalance

The data analyzed for this study is entirely skewed in favor of non-fraudulent transactionsFig. 13. Hence random oversampling is used in an attempt to address the unbalanced data set, however this resulted in over fitting. After attempting SMOTE, the results showed a rise in fraudulent transactions from 15percentage to 65percentage and the non-fraudulent transaction went to 35percentage in a way the data set is now biased towards the fraud transaction.

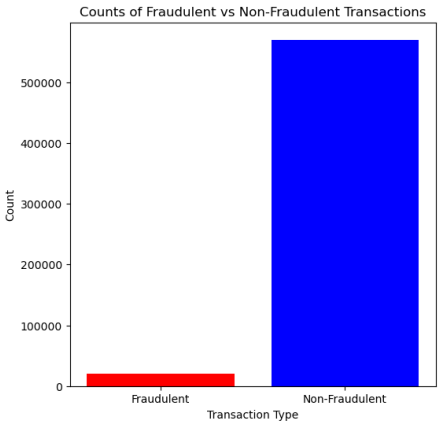

Figure 13: Imbalanced data

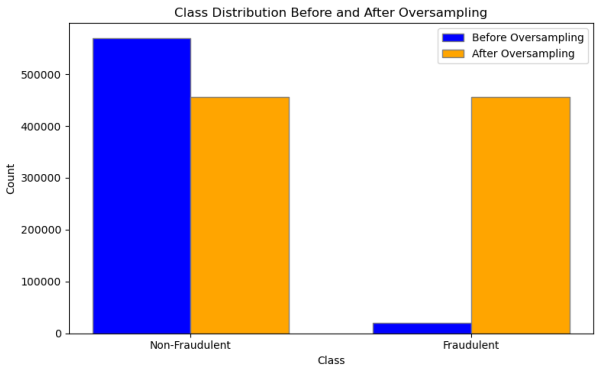

Figure 14: Hybrid - sampling

Furthermore, over-sampling could identify a large proportion of frauds, but it may result in a small rise in reporting normal transactions as fraudulent, according to a research on credit card fraud detection using the SMOTE approachMeng et al. (2020). This demonstrates the balance between reducing false positives and effectively identifying fraudulent transactions. To overcome this, hybrid approaches are utilized, which mix under sampling and oversampling techniques and may be useful in effectively resolving the issue of class imbalance. Here, firstly the SMOTE is applied on the training set of data, and then Random Under sampling is utilized to the synthetic data that SMOTE produced. This produce a balanced data set with 50percentage fraudulent and 50percentage non-fraudulent transactions Fig. 14.

3.2.3 Outlier check

Handling outliers in a data set is essential, particularly when it comes to fraud detection where the sample are completely unbalanced. Z-score and DBSCAN were the two outlier

identification techniques used in this case study to deal with the high number of outliers.

Z-score Method entails figuring out each data point's Z-score and eliminating any that don't meet a certain threshold. In this instance, the data set size was significantly reduced from 590,540 to 394,344 using the Z-score approach. But all of the fraudulent transactions were eliminated, leaving a data set that contained only legitimate transactions.

DBSCAN Method: An outlier detection clustering approach is the density-based spatial clustering of applications with noise, or DBSCAN.The data set size was further reduced to 50,906 by using the DBSCAN approach, and there were 1,209 fewer fraudulent transactions overall. It was decided to move forward without outlier removal due to the significant decrease in the number of fraudulent transactions following outlier removal. The large imbalance in the data set fraudulent transactions made up just 3.5percentage of all transactions likely had an impact on this choice.

3.3 Data Preparation

This section provides with the aim of preparing the data for machine learning by addressing the pre-processing results and carrying out the subsequent tasks:

3.3.1 Detail of Target Variable

This real IEEE-CIS Fraud Detection data set contains a binary variable 'isFraud' which is the target variable that indicates whether a transaction is fraudulent (isFraud=1) or legitimate (isFraud=0). This labeling logic for the 'isFraud' variable is based on reported charge backs on the card, and transactions posterior to it with either user account, email address, or billing address directly linked to these features are also labeled as fraud. If none of the above is reported and found beyond 120 days, then the transaction is defined as genuine. Additionally, there are missing values in the data set, which might be linked to machine or human error or to respondents refusals to provide information on particular topics.

Since this a real data set below things are also to be considered as an insights while handling this data set: In the real world, fraudulent activity might not be reported, e.g., the cardholder was unaware or forgot to report in time, and beyond the claim period. In such cases, supposed fraud might be labeled as legitimate, but it is not possible to know about them. The labeling logic is a complicated situation, and usually, transactions will be flagged as fraud. However, there are exceptions, such as when the billing address was found to be fraudulent in a past transaction because the credit card associated with it was stolen, but the cardholder is actually the victim. In such cases, the cardholder will not be blacklisted forever if they use another legitimate card for future transactions.

In conclusion, the IEEE-CIS Fraud Detection data set relies heavily on the 'isFraud' variable, and there is a chance that choices made regarding data preparation will affect how well fraud detection models work.

3.3.2 Feature Engineering

Based on the above analysis and insights the below feature engineering are done,

1. Analysing the TransactionDT and Card 1-6: By analysing the these features a new feature '**Transaction Frequency**' is created. 'TransactionDT': This column represents

the time stamp of each transaction, its transformed into date-time format, and the difference in days between each transaction for all card columns is used to determine the frequency of transactions for each card identification. 'TransactionAmt': The transaction payment amount is represented in USD, it provides context for understanding the financial implications of the transactions. 'card': column contains the card information, such as card type, card category, issue bank, and country. calculating the transaction frequency, help in identifying patterns and potential fraud activities.

The negative transaction frequency numbers is also investigated which is crucial to ensure the integrity and quality of the data to avoid misleading results and inaccurate predictions however, none of the card identities had any negative transaction frequency values.

2.The new features '**TransactionHour**' and '**TransactionWeekday**' derived from the 'TransactionDT' column provide valuable information related to the timing of the transactions, to identify patterns and potential fraud activities.Fig. 15
TransactionHour: Represents the hour of the day in which the transaction occurred.
TransactionWeekday: Indicates the day of the week, with 0 being Monday and 6 being Sunday.

TransactionAmountLog	TransactionHour	TransactionWeekday	EmailMatch	GeoFeatureEncoded
4.241327	0	2	False	21966
3.401197	0	2	False	27442
4.094345	0	2	False	28558
3.931826	0	2	False	35224
3.931826	0	2	False	33669

Figure 15: Feature Engineering

3.The new feature '**EmailMatch**' is derived using the P and R email domain columns. Fig. 15 The '**EmailMatch**', a new feature, has been created to determine if the recipient's and buyer's email domains are the same or different. 'EmailMatch' helps to calculate the fraud rate to determine whether this new feature may be used to anticipate fraud. The fraud rate for transactions where the email domains match is approximately 9.65percentage, which is significantly higher than the 2.21 percentage fraud rate for transactions where the email domains do not match. Hence, EmailMatch feature could potentially be useful for predicting fraudulent transactions, as there is a notable difference in fraud rates between matched and unmatched email domains.

4. Feature engineering with geographical information using the 'addr1' and 'addr2' columns. A new feature '**GeoFeature**' is created by combining 'addr1' and 'addr2'. Fig. 15 The new encoded geographical feature can be used to improve the performance of fraud detection models by incorporating the geographical information into the analysis

5. Encoding Categorical Value Here a unique number (starting from 0) is assigned to each categorical class of data. In the IEEE data set, below are the categorical columns 'ProductCD', 'card4', 'card6', 'P emaildomain', 'R emaildomain', 'M1', 'M2', 'M3', 'M4', 'M5', 'M6', 'M7', 'M8', and 'M9' and these are being encoded using Label Encoder by replacing with their corresponding numerical labels, allowing models to work with the data more effectively.

3.3.3 Feature Selection

1. Wrapper-based subset feature selection: This technique assesses a subset of features performance which conducts a search for a potential feature from the subset using the induction algorithm. By applying these to every subset and using search strategies to explore the feature combinations that are possible, wrappers evaluate the performance of subsets individually.

2. Lasso Regression: This method performs feature selection by shrinking the coefficients of less important features towards zero Mo et al. (2018).

3. Light gbm - cross validation: This method used to select the best features that help predict the target variables. LightGBM and nested cross-validation to evaluate the model's performance and select the top features using permutation importance i.e. the outer loop performs k-fold cross-validation, and the inner loop performs feature selection.

Feature Reduction method to avoid multi collinearity were also utilized in feature selection process which further reduced the columns to 102. In summary, the heuristic method of correlation matrix and Lasso Regression provided similar kind of results, with a few differences in the selected features. Light GBM's cross-validation has produced a different set of selected features.

3.4 Model Selection Approach

3.4.1 Batch Machine Learning Models

For batch learning, below models are selected:

1. Random Forest: As an ensemble model, final prediction is created by combining the values of the decision trees that it creates. Though it can't achieve state of the art performance for some machine learning tasks, it is often quicker to train and more adaptable than the gradient boosting.Mohamad Aburbeian and Ashqar (2023)

2. Gradient Boosting (LightGBM):It creates a series of decision trees, each of which solves the errors of the one before it. LightGBM accelerates the training process using a histogram-based technique. Huang (2020)

3. Ensemble method (AdaBoost classifier): Sohony et al. (2018) The boosting ensemble model AdaBoost performs especially well when used to decision trees. It increases the weight of data points that were incorrectly categorized in order to learn from the errors made by the prior model.

The reason for choosing these models over others are:
These models efficiently handle the larger data set which is the IEEE data set.
These are resistant to over fitting, which is crucial for preventing the model from capturing the training data patterns too closely and resulting in poor prediction on unseen data.

3.4.2 Online Machine Learning Models

For online learning, below models are selected:

1. Online Random Forest: These adapt to changing data distributions, making them a suitable choice for a wide range of applications and have high accuracy depending on the problem selected. They may create decision trees over time to adjust to changing data

distributions, which makes them highly useful in situations when the data is changing rapidly or is too large to be kept in memory.

2. Online XGBoost: A advanced version of the XGBoost algorithm, Hajek et al. (2022) which is efficient in handling data and a powerful tree-based learning technique, is online XGBoost. These algorithms are suitable for applications in real life where data is continually created since they are designed to manage streaming data.

The reason for choosing these models are:
These models are designed to learn and adapt to new data.
Online Random Forest adapt to new data continuously and handles the dynamic nature of fraud detection patterns. On the other hand, Online XGBoost is an ensemble machine learning algorithm that builds trees sequentially, focusing on the errors made by previous trees.

3.5 Evaluation Metrics

For the comparative study on this research work the below metrics were used Fig. 16:
Accuracy: Its a performance metric that measures the proportion of correct predictions made by a classification model.Naidu et al. (2023)
Recall (True Positive Rate): The proportion of actual fraudulent transactions that are correctly identified as fraudulent.Naidu et al. (2023)

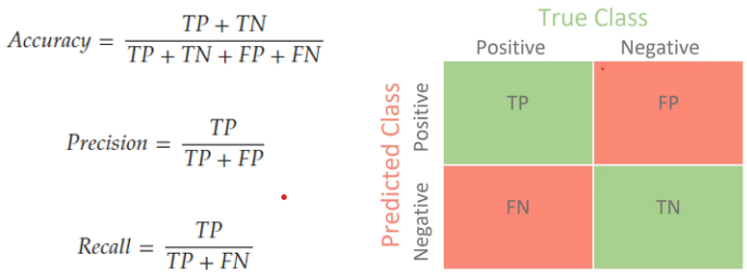

Figure 16: Evaluation Metrics

Precision (Positive Predictive Value): The proportion of predicted fraudulent transactions that are actually fraudulent.Naidu et al. (2023)
F1 Score: The harmonic mean of Precision and Recall, providing a balance between the two metrics.Naidu et al. (2023) F1 = 2 × Precision × Recall / Precision + Recall
Confusion matrix In the context of fraud detection, it compares the predicted fraudulent transactions (positive class) and actual fraudulent transactions (negative class).

4 Design Specification

An approach of detecting fraudulent transactions using real credit card transaction is outlined in the designFig. 17. The system analyses transactions for possible fraud using both online and batch machine learning methods and compares the model performance.

Data Flow Processing: Real Credit Card Data: The data set of credit card transactions is the primary input for the fraud detection process.
Data Pre-Processing: The raw transaction data undergoes pre-processing to clean and

normalize the data, making it suitable for analysis. This step may include handling missing values, encoding categorical variables, and scaling numerical features.
Data Preparation: Further data transformation is performed, this step is crucial for identifying the most relevant features that contribute to the prediction of fraudulent transactions.

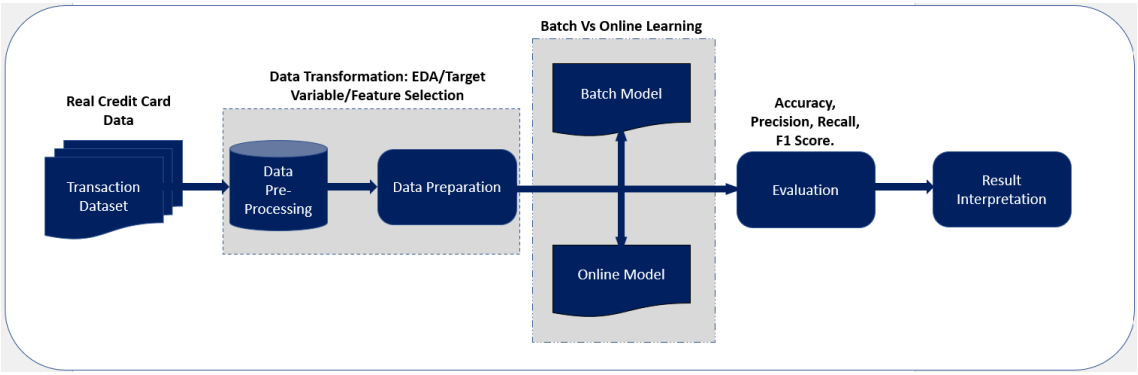

Figure 17: Batch vs Online Machine learning methodology

Model Training and Evaluation:

The system compares two approaches to learning: Batch Models are trained on a fixed set of data and is typically updated at regular intervals. Online Model updates continuously as new data arrives, learning incrementally from each transaction. Here the data set for online machine learning is not streamed instead the test data set is split into batches each 1000 rows and feed that into the model to make it look like the online stream with new data set.
Evaluation: Both models are evaluated on their performance using metrics such as Accuracy, Precision, Recall, and F1 Score. These metrics provide insight into the effectiveness of each model in detecting fraudulent transactions.
Result Interpretation : The final step involves interpreting and comparing the results from the evaluation phase to make informed decisions about which model performs better and how to improve the system's fraud detection capabilities.

5 Implementation

In the IEEE-CIS Fraud Detection data set, Out of the 590,540 card transactions 20,663 (3.5 percentage) are fraudulent. The wrapper subset selection was used to reducing the dimension in the data set. Multiple imputation was used to handle the MNAR missing pattern in the data set. For imbalanced data set handling, SMOTE and Random Under sampling were combined using the Hybrid method. The selected features were scaled and used to train and evaluate machine learning models, including Random Forest, Gradient Boosting (Light GBM), and Ensemble methods (Ada Boost classifier) for batch learning, and Online Random Forest and Online XG Boost for online learning. The evaluation metrics accuracy, recall, precision, PR-AUC curve, and F1score were used to evaluate the models' performance.
The implementation was done using Jupyter Notebook version 6.5.2 and Python 3.11.1. Google Colab cloud was used via the browser for multiple imputation, the run time type is set to Python 3, and the hardware accelerator was T4GPU. The system was implemented

on a Windows 11 machine with an AMD Ryzen 3 5300U processor, 24GB RAM, and 500GB storage. The research project is expected to produce a thorough analysis of the data set, a comparison of the model's performances, insights into the performance of the suggested fraud detection system, and suggestions for additional features and improvements based on findings.

6 Evaluation and Results

The research work compared the performance of various machine learning models for fraud detection, including Random Forest, Gradient Boosting (LightGBM), Ensemble Adaboost, Online Random Forest, and Online XGBoost.

6.1 Experiment 1: Batch Machine learning Model

6.1.1 Random Forest

The Random Forest classifier performed well overall, with high accuracy (97.41) and precision (72.67). However, the recall rate (43.47) suggests a balance between accurately detecting positive instances and reducing false negatives. The models attention to recall is shown in F1 score 51.68. PR-AUC score of 59.59, The curve starts with a high precision at low recall values, when predicting predicts a positive class model is highly confident about its prediction. As recall increases, precision gradually decreases. This is expected because as the classifier attempts to capture more true positives making more false positive errors, thus reducing precision. confusion matrix: 113,207 true negatives, 659 false positives, 2,398 false negatives, and 1,844 true positives.

6.1.2 Gradient Boosting (Light GBM)

High accuracy (92.74) and F1 score (30.4) shows the balance between precision and recall. On the other hand, the recall (44.31) indicates a moderate ability to identify true positives and precision (23.23) suggests a high rate of false positives. PR-AUC score of 34.77, indicates moderate precision-recall performance. The curve starts at a high precision value, indicating that when the recall is low, the precision is high. Given that the value is less than 0.5, shows that accuracy and recall are balanced, but it also signals that the classifier may need to be improved. Confusion matrix: 107,656 true negatives, 6,210 false positives, 2,362 false negatives, and 1,880 true positives.

6.1.3 Ensemble (Adaboost)

While the accuracy of the AdaBoost classifier was 86.85, it performed less well than previous models in terms of recall (48.53) and precision (13.38). The focus on recall is shown by the F1 score of 20.96. PR-AUC score of 31.87 As recall increases, precision declines this indicates that as the classifier becomes less selective to include more true positives, it also includes more false positives, thus reducing precision. Shape of the curve suggests that the classifier struggles to maintain a high precision as it tries to increase recall but consistent with the lower precision score. Confusion matrix : 100,527 true negatives, 13,339 false negatives, 2,183 false positives, and 2,059 true positives.

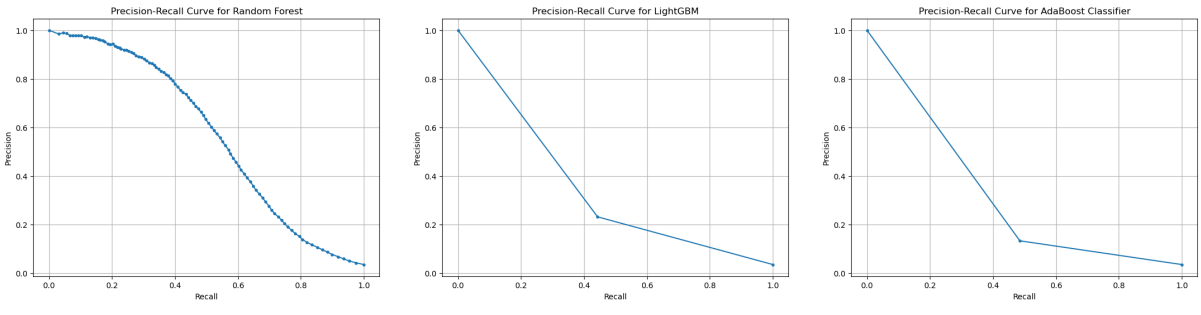

Figure 18: Batch Machine Learning

6.2 Experiment : Online Machine learning Model

6.2.1 Online Random Forest

The Online Random Forest model provided results with a high accuracy of 97.42 and precision of 73.50 shows the model capability to make accurate positive predictions. On the other hand, the average recall (43.00) points to a compromise between minimising false negatives and accurately detecting positive instances. Recall and accuracy are balanced in the model, as seen by the F1 score of 53.89. There are 2,632 false negatives, 1,848 true positives, 767 false positives, and 112,861 true negatives in the confusion matrix.Fig. 19

6.2.2 Online XGBoost Classifier

A good overall performance was shown by the Online XGBoost classifier, which attained high average accuracy (95.94) and precision (42.91). On the other hand, the average recall (42.20) indicates an imbalance between minimising false negatives and accurately detecting positive instances. The models 42.24 F1 score shows the recall and accuracy are less balanced. There are 111,507 true negatives, 2,359 false positives, 2,436 false negatives, and 1,806 true positives in the combined confusion matrix.Fig. 20

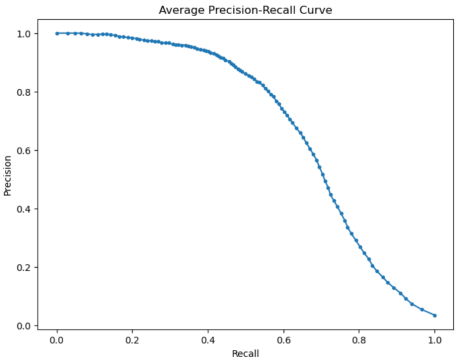

Figure 19: Online Random Forest

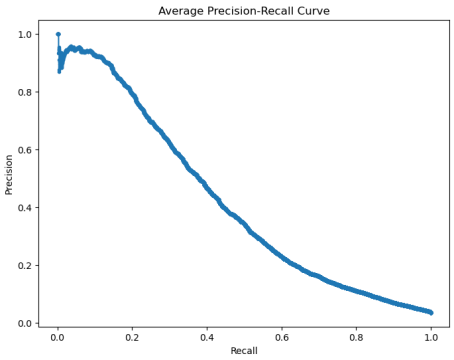

Figure 20: Online XGBoost

7 Discussion

The Random Forest batch model performs better than the online machine learning models when it relates to fraud detection. It has the greatest accuracy of 97.41 and a PR-AUC of 59.58, which shows that it can effectively identify between fraudulent and non-

fraudulent transactions. Another batch model, LightGBM, has relatively small scores for all metrics and a balanced profile, but its precision is much lower at 23.68. AdaBoost has a competitive recall of 48.53, indicating that it is more capable at finding positive cases, despite its lower accuracy of 86.85. Even though the online models are not as precise as the best batch model, they still work well. With a precision of 73.5 and an accuracy of 97.42, the Online Random Forest performs almost identically to its batch counterpart. With the recall and F1 score compared to any model 43.01 and 53.89, respectively and an accuracy of 95.95, online XGBoost shows how effective it is in detecting fraudulent cases a crucial aspect of fraud detection.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**PR-AUC**
Random Forest	97.41	72.67	43.47	51.68	59.59
AdaBoost	86.86	13.37	48.54	20.97	31.88
LightGBM	92.74	23.24	44.32	30.49	34.78
Online Random Forest	97.42	73.50	43.01	53.89	56.83
Online XGBoost	95.95	42.92	42.21	42.24	42.05

Table 2: Model Performance Metrics

In summary, the Online Random model balances higher precision, recall and F1 score, making it highly useful for fraud detection showing their capacity to detect fraudulent transactions with minimal false positives and false negatives. When comparing the batch and online models, the online Random Forest model emerges as the best choice in this study, with a slightly higher accuracy and a comparable F1 score to the batch version. For the reason that Online Random forest model seems to be the best algorithm out of the five because of its strong performance in detecting fraud with high accuracy.

8 Conclusion and Future Work

The research evaluated multiple machine learning models for fraud detection using distinct evaluation metrics. The findings showcase detailed differences in performance across various classifiers (Batch and Online models), each highlighting different aspects of model behavior.

Insights and Recommendations: The Online Random Forest classifier emerged as the most balanced model, delivering strong overall accuracy while maintaining a balance between precision and recall. However, for cases prioritizing the identification of fraudulent transactions, models like Adaboost might be preferable due to their higher recall rates. Considering the nature of fraud detection where false negatives (undetected fraud) can be critical the Recall metric holds significant importance. Although the Adaboost model excelled in recall, its poor precision raises concerns about false positives. Further fine-tuning of models could enhance precision without compromising recall, aiming for a more balanced detection approach.

Future Work: In real-world applications, a hybrid approach combining the strengths of Random Forest for overall accuracy and Adaboost for better fraud identification might offer a comprehensive solution. However, continued research and model refinement are essential for optimal fraud detection while minimizing false positives. The choice of a suitable model must align with the specific needs and priorities of the application, balancing between overall accuracy and the critical importance of detecting fraudulent transactions.

References

Baabdullah, T., Rawat, D. B., Liu, C. and Alzahrani, A. (2022). An ensemble-based machine learning for predicting fraud of credit card transactions, *Science and Information Conference*, Springer, pp. 214–229.

Bisong, E. and Bisong, E. (2019). Batch vs. online learning, *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* pp. 199–201.

Burlutskiy, N., Petridis, M., Fish, A., Chernov, A. and Ali, N. (2016). An investigation on online versus batch learning in predicting user behaviour, *Research and Development in Intelligent Systems XXXIII: Incorporating Applications and Innovations in Intelligent Systems XXIV 33*, Springer, pp. 135–149.

Chawla, N. V., Bowyer, K. W., Hall, L. O. and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique, *Journal of artificial intelligence research* **16**: 321–357.

Desuky, A. S. and Hussain, S. (2021). An improved hybrid approach for handling class imbalance problem, *Arabian Journal for Science and Engineering* **46**: 3853–3864.

Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M. and Alam, S. (2023). Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques, *Procedia Computer Science* **218**: 2575–2584.

Hajek, P., Abedin, M. Z. and Sivarajah, U. (2022). Fraud detection in mobile payment systems using an xgboost-based framework, *Information Systems Frontiers* pp. 1–19.

Hasan, M. K., Alam, M. A., Roy, S., Dutta, A., Jawad, M. T. and Das, S. (2021). Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021), *Informatics in Medicine Unlocked* **27**: 100799.

Heymans, M. W. and Twisk, J. W. (2022). Handling missing data in clinical research, *Journal of clinical epidemiology* **151**: 185–188.

Hoi, S. C., Sahoo, D., Lu, J. and Zhao, P. (2021). Online learning: A comprehensive survey, *Neurocomputing* **459**: 249–289.

Huang, K. (2020). An optimized lightgbm model for fraud detection, *Journal of Physics: Conference Series*, Vol. 1651, IOP Publishing, p. 012111.

Kang, H. (2013). The prevention and handling of the missing data, *Korean journal of anesthesiology* **64**(5): 402–406.

Lee, K. J., Carlin, J. B., Simpson, J. A. and Moreno-Betancur, M. (2023). Assumptions and analysis planning in studies with missing data in multiple variables: moving beyond the mcar/mar/mnar classification, *International Journal of Epidemiology* p. dyad008.

Liu, C.-H., Tsai, C.-F., Sue, K.-L. and Huang, M.-W. (2020). The feature selection effect on missing value imputation of medical datasets, *applied sciences* **10**(7): 2344.

Meng, C., Zhou, L. and Liu, B. (2020). A case study in credit fraud detection with smote and xgboost, *Journal of Physics: Conference Series*, Vol. 1601, IOP Publishing, p. 052016.

Mo, J., Deng, Z., Jia, B., Jiang, H. and Bian, X. (2018). A novel fll-assisted pll with fuzzy control for tc-ofdm carrier signal tracking, *IEEE Access* **6**: 52447–52459.

Mohamad Aburbeian, A. and Ashqar, H. I. (2023). Credit card fraud detection using enhanced random forest classifier for imbalanced data, *arXiv e-prints* pp. arXiv–2303.

Naidu, G., Zuva, T. and Sibanda, E. M. (2023). A review of evaluation metrics in machine learning algorithms, *Computer Science On-line Conference*, Springer, pp. 15–25.

Ramani, K., Suneetha, I., Pushpalatha, N. and Harish, P. (2022). Gradient boosting techniques for credit card fraud detection, *Journal of Algebraic Statistics* **13**(3): 553–558.

Sadgali, I., Sael, N. and Benabbou, F. (2019). Performance of machine learning techniques in the detection of financial frauds, *Procedia computer science* **148**: 45–54.

Sohony, I., Pratap, R. and Nambiar, U. (2018). Ensemble learning for credit card fraud detection, *Proceedings of the ACM India joint international conference on data science and management of data*, pp. 289–294.

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. and Kuruwitaarachchi, N. (2019). Real-time credit card fraud detection using machine learning, *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, pp. 488–493.