**Transactional Fraud Detection**

A

Minor Project Report

#### Submitted by

##### Anukriti Manchanda 01096402720

##### Anant Parashar 00796402720

##### Shubham Raturi 03796402720

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE & ENGINEERING

#### Under the Guidance of

Mr. Ajay Kumar Tiwari (Assistant Professor,CSE)



# MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

Department of Computer Science & Engineering



# CERTIFICATE

This is to Certified that this MINOR project report “**Transactional Fraud Detection**” is submitted by “**Anukriti Manchanda** (01096402720), **Anant Parashar** (00796402720), **Shubham Raturi** (03796402720)” who carried out the project work under my supervision. I approve this MINOR project for submission.

Prof. Namita Gupta Mr. Ajay Kumar Tiwari (HoD, CSE) (Guide)

**ABSTRACT**

In the ever-evolving financial landscape, the prevalence of fraudulent activities has given rise to substantial financial losses. Recognizing the urgency to combat this menace, we have adopted an aggressive and prediction-centered approach to accurately identify and halt fraudulent transactions, safeguarding against potential widespread losses. This project articulates a comprehensive data science methodology aimed at constructing a predictive model, delving into the intricate nuances of fraud detection within the realm of mobile transactions. By leveraging advanced predictive modeling, we strive to fortify our defenses against the dynamic tactics employed by fraudsters in the financial domain.

This project addresses the growing risk of financial fraud with increased security investments, striking a delicate balance between aggressive customer acquisition and the risks associated with potential fraud failures. The overall goal is to develop a machine learning model that not only maximizes prediction accuracy, but also strategically minimizes false positives and provides a balanced and effective approach to fraud prevention. Presented as a ten-step data science initiative, the solution includes extensive data collection, feature planning, exploratory data analysis, and culminates in the deployment of a robust predictive model via the Flask API. Significant data challenges prevailing assumptions, revealing that any fraudulent transaction exceeding the threshold of R$ 10,000 is a key indicator of suspicion. The project and its significance is that it can increase the income of Blocker Fraud Company and contribute significantly to the wider financial transaction fraud prevention landscape, placing the company at the forefront of the industry with its innovative payment structure and reliable forecast. model Future considerations include continuous adaptation of the model, integration of new technologies, and research of the application of similar models in different financial contexts, pointing to a future where advanced predictive models become an integral part of the wider spectrum of financial collateral..

**ACKNOWLEDGEMENT**

It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Mr. Ajay Kumar Tiwari (Assistant Professor, CSE) MAIT Delhi, for theirvaluable guidance, encouragement and help for completing this work. Their useful suggestions for this whole work and co-operative behavior are sincerely acknowledged.

I also wish to express my indebtedness to my parents as well as my family member whose blessings and support always helped me to face the challenges ahead.

|  |  |  |
| --- | --- | --- |
| Anukriti Manchanda | Anant Parashar | Shubham Raturi |
| 01096402720 | 00796402720 | 03796402720 |

Place: Delhi Date:

### TABLE OF CONTENTS

**Page No.**

[Certificate 1](#_bookmark0)

[Abstract 2](#_bookmark1)

Acknowledgements 3

Table of Content 4

[List of Tables 5](#_bookmark2)

List of Symbols, Abbreviations and Nomenclature 6

[CHAPTER 1 INTRODUCTION 7](#_bookmark3)

CHAPTER 2 LITERTATURE SURVEY 10

[CHAPTER 3 RESEARCH / APPROACH 16](#_bookmark4)

[CHAPTER 4 RESULTS 23](#_bookmark5)

[CHAPTER 5 CONCLUSION 26](#_bookmark6)

[CHAPTER 6 SUMMARY 27](#_bookmark7)

[CHAPTER 7 FUTURE SCOPE 28](#_bookmark8)

[CHAPTER 8 REFERENCES 29](#_bookmark9)

CODE 31

[RESEARCH PAPER 41](#_bookmark10)

PUBLICATION PROOF 46

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **Name** | **Page No.** |
| 1. | Tech Stack Used | 16 |

**LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURES**

|  |  |  |
| --- | --- | --- |
| **S NO** | **Abbreviation** | **Definition** |
| 1. | AI | Artificial Intelligence |
| 2. | ML | Machine Learning |
| 3. | KNN | K Nearest Neighbors |
| 4. | SVM | Support Vector Machine |
| 5. | NN | Neural Network |

## CHAPTER 1: INTRODUCTION

#### BACKGROUND

In In modern financial transactions, the emergence of digital technologies has brought with it unprecedented convenience and efficiency, but has also opened financial systems to an increasing risk - transaction fraud. As financial transactions increasingly move to online platforms, the complexity of electronic exchanges has become a breeding ground for sophisticated fraud. The complexity and frequency of these illegal schemes have surpassed traditional rules-based detection systems, necessitating innovative approaches to protect the integrity of financial ecosystems.

According to the Reserve Bank of India (RBI), there were 13,530 bank fraud cases in India in the financial year 2023.

With the emergence of advanced technologies, the realm of digital transactions has witnessed a paradigm shift, paving the way for innovative solutions to detect fraud transactions.

However, evolving tactics and vast data complexities pose challenges in effective transactional fraud detection systems. Recognizing this challenge, this project addresses the urgent demand for advanced transactional fraud detection mechanisms by immersing itself in the world of advanced technologies. Traditional methods that were once effective now struggle to cope with the evolving tactics used by fraudsters to exploit vulnerabilities in electronic transactions. The core of this initiative is to create a robust framework for transactional fraud detection, primarily using machine learning and data analytics capabilities. Known for their adaptability and ability to discern patterns in large data sets, these technologies offer a promising means of fortifying financial systems against emerging threats.

In the context of the growing financial landscape, this project is based on the ability to understand the challenges that companies face in navigating the complexities of fraud detection for transactions, and to rethink and streamline the fraud detection process using new technologies.

#### PROBLEM STATEMENT

The growth of e-commerce has led to an increase in credit card frauds, causing significant losses for both customers and businesses. To reduce these losses, an effective fraud discovery system is needed. Research has focused on various models and styles, but some datasets face imbalances. Fraud discovery can be divided into anomaly and abuse discovery, with anomaly systems using supervised literacy and abuse systems using unsupervised literacy. Frauds can be classified into traditional card-associated, trafficker- related, and internet frauds.

Transactional fraud detection tools lack personalization and targeted guidance, relying on generic algorithms. This one-size-fits-all approach fails to address diverse needs of institutions, requiring a tailored, adaptive solution that considers individual financial entities.

This project emerges as a response to this prevailing problem.

#### FEASIBLITY STUDY

It was essential to carry out a feasibility study in order to evaluate the viability and possible success of the suggested solution before to starting the project.

**Technical Feasibility**: It hinges on evaluating the availability and quality of transactional data, assessing the adaptability of existing technology infrastructure for advanced fraud detection tools, and ensuring the scalability of the system to handle increasing transaction volumes without performance compromise..

**Financial Feasibility**: Financial feasibility entails a detailed cost-benefit analysis, evaluating the financial viability of implementing a fraud detection system against potential fraud losses. ROI calculation considers setup costs and ongoing maintenance expenses, ensuring informed decision-making.

**Operational Feasibility**: Operational feasibility for a fraud detection system centers on assessing user acceptance, seamless integration with existing processes, and identifying training needs for personnel. This comprehensive evaluation ensures a smooth assimilation into workflows and effective system utilization.

#### OBJECTIVE

* + - The primary objective of transactional fraud detection is to identify and prevent illicit activities within financial transactions. This encompasses unauthorized access, deceitful transactions, and various forms of fraudulent behavior.
    - By leveraging advanced technologies such as machine learning and data analytics, the objective is to develop a proactive and adaptive system that can swiftly detect anomalies, patterns, and potential fraud indicators in real-time, mitigating risks and enhancing the overall security of electronic transactions..

The goal is to provide an advanced fraud detection system that provides real-time information and relevance to detect anomalies and potential fraud. This approach involves adapting the system to the specifics of each financial entity, promoting more accurate and effective protection against changing fraud tactics. The goal is to strengthen financial systems by implementing a dynamic and adaptive transactional fraud detection framework that meets the unique challenges presented by the ever-changing landscape of fraudulent activity.

#### ADVANTAGES

**Early Fraud Identification**: Transactional fraud detection allows for the early identification of fraudulent activities, enabling prompt intervention and mitigation to minimize potential financial losses.

**Enhanced Security**: By leveraging advanced technologies such as machine learning and data analytics, transactional fraud detection systems provide a robust and adaptive security framework, staying ahead of evolving fraudulent tactics and ensuring the integrity of financial transactions.

**Cost Savings**: Detecting and preventing fraudulent transactions early can lead to substantial cost savings for individuals and organizations, preventing financial losses and the need for extensive recovery efforts.

**Data-Driven Insights**: The analysis of transactional data provides valuable insights into patterns and trends, empowering institutions with data-driven decision-making capabilities for continuous improvement of fraud prevention strategies.

## CHAPTER 2: LITERATURE SURVEY

##### Zareapoor, M., Seeja.K.R, S. K. R., & Afshar Alam, M. (2012). Analysis on credit card fraud detection techniques based on certain design criteria :

This paper discusses the increasing problem of credit card fraud and the need for fraud detection systems to minimize losses for credit card issuing banks. The most commonly used fraud detection methods are Neural Network (NN), rule-induction techniques, fuzzy system, decision trees, Support Vector Machines (SVM), Artificial Immune System (AIS), genetic algorithms, and K-Nearest Neighbor algorithms. These techniques can be used alone or in collaboration using ensemble or meta-learning techniques to build classifiers. The paper presents a survey of these techniques and evaluates each methodology based on certain design criteria such as speed, accuracy, and cost. The results show that different techniques have different strengths and weaknesses. For example, the Fuzzy Darwinian system has a very high accuracy but very low processing speed, while Hidden Markov Model has a fast processing speed but low accuracy. Bayesian Network is very high in speed processing with good accuracy, and Decision Tree has a very fast processing speed. The paper concludes that a hybrid approach combining different techniques may be the most effective way to detect credit card fraud with higher accuracy. The document provides a comprehensive overview of credit card fraud detection techniques and their strengths and weaknesses, making it a valuable resource for researchers and practitioners in the field.

##### Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression :

The increasing number of credit card transactions has led to a rise in fraud cases, and dealing with noisy and imbalanced data has made it difficult to detect fraud. The proposed system uses logistic regression as a classifier and incorporates a pre-processing step to handle dirty data. Two methods, mean-based and clustering-based, are used to clean the data. The proposed classifier outperforms two well-known classifiers, support vector machine and voting classifier, in terms of accuracy, sensitivity, and error rate. The document also highlights the motivation behind the research, the problem statement, the contributions of the proposed system, and the structure of the paper. It concludes by discussing future work and limitations of the proposed classifier.

1. **Saheed, Y. K., Hambali, M. A., Arowolo, M. O., & Olasupo, Y. A. (2020). Application of feature selection on Naive Bayes, random forest and SVM for credit card fraud detection :** In this paper the use of genetic algorithms for feature selection in credit card fraud detection at the application level is discussed. The authors apply three different supervised machine learning techniques to the German credit card dataset to test the effectiveness of their approach. They find that the first priority features selected by the genetic algorithm are the most important for detecting fraud, and that the Random Forest algorithm outperforms Naïve Bayes and Support Vector Machine in terms of accuracy, fraud detection rate, and precision. The paper provides a detailed framework for the proposed system and includes tables and formulas to support the experimental findings.

##### D. Tanouz, R. R. Subramanian, D. Eswar, G. V. P. Reddy, A. R. Kumar and C. V. N. M. Praneeth, Credit Card Fraud Detection Using Machine Learning, *2021 :*

It explores different machine learning algorithms that can be used to detect and prevent fraudulent activities in credit card transactions. The document emphasizes the importance of data mining techniques, such as under-sampling and outlier detection, in improving the accuracy of fraud detection. The paper also discusses the results of experiments conducted to measure the accuracy, recall, precision score, and F1 score of different algorithms. The authors conclude that data mining techniques are essential for detecting fraudulent activities accurately, and machine learning algorithms can help prevent financial losses to banks and customers. Overall, it provides valuable insights into the use of machine learning for credit card fraud detection.

##### Mosa M. M. Megdad, Samy S. Abu-Naser & Bassem S. Abu-Nasser. Fraudulent Financial Transactions Detection Using Machine Learning, 2022 :

They presents a study on detecting fraudulent financial transactions using machine learning algorithms. The authors collected a dataset of fraudulent transactions from Kaggle Depository, which consists of 6362620 records with 10 features. They explore different methods to tackle the issue of imbalanced classification and find that the existing methods are costly and show many false alarms. The authors also provide a summarized result and weakness of using a credit card fraud labeled dataset. The study concludes that machine learning algorithms can be used to predict the legitimacy of financial transactions and minimize financial loss. However, the findings of this study may not be applicable to other financial companies and industries.

##### Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, J. Christopher Westland. Data mining for credit card fraud: A comparative study, 2022 :

A study on using data mining techniques for credit card fraud detection. They compare the performance of three different approaches: logistic regression, random forests, and support vector machines. And then evaluated the models using various measures, including accuracy, sensitivity, specificity, and precision. The results show that all three models perform well, but random forests and support vector machines outperform logistic regression. They also suggested that credit card companies could use these models to score transactions and identify potential fraud. Overall, the study highlights the potential benefits of using advanced data mining techniques for credit card fraud detection and provides insights into the strengths and weaknesses of different approaches.

##### Siddharth Shinde, Pravin Chavan, Dr. Mrs. K. S. Tiwari. Credit Card Fraud Detection: A Comparative Study, 2021 :

It discusses the use of supervised machine learning algorithms to detect credit card fraud. They apply various algorithms to a real-world dataset and identify the most important variables for accurate detection. They also implement a super classifier using ensemble learning methods to improve accuracy. The credit card dataset is highly imbalanced, with more legitimate transactions than fraudulent ones. The paper compares the performance of various supervised machine learning algorithms against the super classifier. The authors use accuracy, F1-Score, Recall, Precision, G-Mean, FPR, TRP, and specificity to compare the models. The study highlights the importance of data analytics in detecting hidden patterns and making informed decisions in situations such as credit card fraud.

##### F. N. Ogwueleka, “Data Mining Application in Credit Card Fraud Detection System,” vol. 6, no. 3, pp. 311–322, 2011 :

They used data mining techniques, specifically the self-organizing map neural network (SOMNN), in credit card fraud detection. The study highlights the importance of timely information on fraudulent activities in the banking industry and the potential for valuable business information to be extracted from large databases. The SOMNN technique was found to be effective in optimal classification of transactions into legitimate and fraudulent groups, with a receiver-operating curve (ROC) detecting over 95% of fraud cases without causing false alarms.

The system also supports different types of alerts with varying priority levels, allowing users to target resources and be informed of specific alert types and levels. The input and output database

interfaces provide a user-friendly way to import transaction data and present detection results. Overall, the study presents promising results for the use of data mining techniques in credit card fraud detection.

##### Leskovec, J., Rajaraman, A., & Ullman, J. (2014). Data Mining In Mining of Massive Datasets (pp.1-18) Cambridge: Cambridge University Press :

The Mining of Massive Datasets is a comprehensive guide to practical algorithms for data mining, covering topics such as map-reduce framework, locality-sensitive hashing, stream processing algorithms, PageRank, frequent itemsets, clustering, recommendation systems, and Web advertising. The book is written by Jure Leskovec, Anand Rajaraman, and Jeff Ullman, and is based on material developed over several years for a one-quarter course at Stanford. The second edition includes new and extended coverage on social networks, machine learning, and dimensionality reduction, making it essential reading for students and practitioners alike. The book provides a thorough introduction to the field of data mining and is a valuable resource for anyone interested in learning about practical algorithms for analyzing large datasets.

##### Hegazy, Mohamed. (2016). Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques :

The proposed algorithm aims to identify unified patterns per customer, including both normal and fraudulent behavior, by introducing the LINGO clustering algorithm. This approach provides a more efficient and accurate way of detecting fraudsters and handles imbalanced datasets. The paper also discusses the methodology and techniques used for implementation and evaluates the performance of the enhanced algorithm with the original fraud miner proposed in a previous study. The proposed method offers a promising approach to credit card fraud detection and could potentially be applied to other types of financial fraud.

##### Sagadevan, Saravanan & Malim, Nurul & Yee, Ong. (2018). Credit Card Fraud Detection Using Machine Learning As Data Mining Technique :

They explores the use of data mining and machine learning techniques to prevent fraudulent activities in the financial industry, specifically in credit card transactions. The research methodology involves using the open-source tool WEKA to measure the performance of classifiers, which are trained and tested using a 10-fold cross-validation technique. The data is transformed and reduced to appropriate forms before being fed into the machine learners. The

paper tests five Bayesian classifiers and evaluates them using two datasets, one of which is a dummy dataset representing credit card data, and the other is a newly transformed dataset using data normalization and Principal Component Analysis techniques. Overall, all the Bayesian classifiers achieved significantly better results after being fed with filtered data. The paper concludes that the combination of data mining and machine learning techniques can accurately distinguish between genuine and non-genuine credit card transactions.

##### K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," in IEEE Access, vol. 6, pp. 14277-14284, 2018 :

They presented a study on the use of machine learning algorithms to detect credit card fraud, which is a major concern in the financial industry. The paper discusses three popular algorithms - Linear Regression, Logistic Regression, and Support Vector Machines - and compares their strengths and limitations. The authors then propose a hybrid approach using AdaBoost and Majority Voting, which combines the strengths of multiple algorithms to improve accuracy and reduce false positives. The study uses a real credit card dataset from a financial institution in Malaysia and achieves high accuracy and sensitivity rates in detecting fraud. The paper also provides a brief biography of the lead author, who is an expert in signal processing and machine learning with over 550 technical publications and numerous awards.

##### Nader Mahmoudi, Ekrem Duman, Detecting credit card fraud by Modified Fisher Discriminant Analysis :

The paper outlines the importance of credit card fraud detection and the challenges associated with it. The authors propose the use of MFDA as a linear discriminant function to classify transactions as fraudulent or legitimate. The paper also discusses related works and the experimental settings used to investigate the proposed methods. The performance of the methods is evaluated using a confusion matrix and classical performance measures. The paper concludes with a discussion of the results and possible future studies. Overall, the paper provides valuable insights into the use of linear discriminant functions in credit card fraud detection and highlights the need for further research in this area.

##### Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, Gianluca Bontempi, Learned lessons in credit card fraud detection from a practitioner perspective :

They provides insights into designing efficient credit card fraud detection algorithms from a practitioner's perspective. The document highlights the challenges of detecting fraud in credit card transactions, including the unbalanced nature of the data and the non-stationary distribution of fraudulent activities. The authors suggest that advanced machine learning techniques, such as Random Forests, can assist fraud investigators in detecting fraudulent transactions. They provides a detailed explanation of the various strategies used in credit card fraud detection, including algorithm selection, sampling methods, model update frequency, and incremental approaches. The authors suggest that the sum of ranks for a given strategy can be used to determine its effectiveness in detecting fraudulent transactions.

## CHAPTER 3: RESEARCH/APPROACH

#### PROBLEM IDENTIFICATION AND CONTEXTUAL RESEARCH

Transactional fraud poses a significant threat to financial institutions, businesses, and consumers worldwide. As electronic transactions continue to rise, so does the sophistication of fraudulent activities. The increasing prevalence of fraud necessitates robust and adaptive fraud detection systems to safeguard financial transactions and maintain trust in digital commerce.

Despite advancements in technology and the application of machine learning in fraud detection, challenges persist. Existing systems may struggle to keep pace with evolving fraud tactics, leading to a growing gap between the methods employed by fraudsters and the detection capabilities of financial institutions. This research aims to address these challenges and enhance the efficacy of transactional fraud detection.

#### Research Context

The research is situated within the broader context of cybersecurity, financial technology (FinTech), and data analytics. With the continuous digitization of financial transactions, the need for proactive and intelligent fraud detection mechanisms becomes paramount. This study focuses on leveraging machine learning techniques for transactional fraud detection, aiming to develop models that not only detect known patterns but also adapt to emerging fraud trends.

#### Significance of the Study

The significance of this study lies in its potential to contribute to the advancement of transactional fraud detection methodologies. By addressing the limitations of existing systems, the research seeks to improve the accuracy and adaptability of fraud detection models. Successfully mitigating fraud risks can have far-reaching impacts, including reducing financial losses for businesses, enhancing consumer trust in digital transactions, and contributing to the overall resilience of financial ecosystems.

#### Review of Existing Literature

A comprehensive review of existing literature reveals the current landscape of transactional fraud detection. Previous studies have explored various approaches, including rule-based

systems, supervised and unsupervised machine learning models, deep learning techniques, and ensemble methods. However, drawbacks such as imbalanced datasets, evolving fraud patterns, and interpretability challenges persist. This study builds upon the insights gained from prior research to develop novel strategies that address these limitations.

#### Research Objectives

The primary objectives of this research are as follows:

* + - 1. To analyze the effectiveness of existing transactional fraud detection systems.
      2. To identify and address the limitations of current methodologies, including challenges related to imbalanced datasets, evolving fraud patterns, and interpretability.
      3. To develop and evaluate novel machine learning models that enhance the accuracy and adaptability of transactional fraud detection.
      4. To contribute valuable insights and practical recommendations for the improvement of real-world fraud detection systems.

In the subsequent sections, we delve into the methodology, data collection, and the proposed models to achieve these objectives.

#### TECHNOLOGY EVALUATION AND SELECTION

|  |  |
| --- | --- |
| **S. NO.** | **TECH STACK USED** |
| 1. | Python |
| 2. | Numpy,Pandas |
| 3. | Seaborn, Matplotlib |
| 4. | Logistic Regression, K Nearest Neighbors |
| 5. | Random Forest, Support Vector Machine,  XGBoost |
| 6. | Python Flask |

*Table 1: Tech Stack Used*

##### PYTHON :

Python is a dominant language for machine learning (ML) due to its simplicity, extensive libraries, and community support. Libraries such as NumPy, Pandas, and Scikit-learn facilitate data preprocessing, analysis, and modeling. Frameworks like TensorFlow and PyTorch provide robust tools for building and training complex neural networks. Python's versatility, readability, and integration capabilities make it a preferred choice for ML practitioners across various domains and skill levels.

##### NUMPY :

NumPy, a fundamental library for machine learning in Python, enables efficient numerical operations and array manipulation. It provides a versatile N-dimensional array object, facilitating mathematical operations and statistical analysis. NumPy's speed and functionality make it a cornerstone in ML workflows, supporting tasks like data preprocessing, feature engineering, and matrix computations, crucial for algorithm implementations and model training.

* + - **PANDAS :**

Pandas is a key library for machine learning in Python, offering powerful data manipulation tools. It provides DataFrame structures for efficient handling of structured data, facilitating tasks like data cleaning, exploration, and transformation. Pandas seamlessly integrates with other ML libraries, such as NumPy and Scikit- learn, making it an essential component for preprocessing and organizing datasets in diverse machine learning projects.

##### SEABORN :

Seaborn, a statistical data visualization library in Python, enhances machine learning (ML) projects by providing an aesthetically pleasing and informative interface for creating insightful plots. Built on Matplotlib, Seaborn simplifies complex visualizations, enabling ML practitioners to explore patterns and relationships in their data effortlessly. Its integration with Pandas makes it a valuable tool for

visualizing datasets and gaining a deeper understanding of the underlying structures before implementing ML algorithms.

##### MATPLOTLIB :

Matplotlib, a versatile data visualization library in Python, plays a crucial role in machine learning projects by creating clear and informative plots. Widely used in tandem with other libraries like NumPy and Pandas, Matplotlib allows ML practitioners to visually analyze data distributions, patterns, and relationships, aiding in decision-making during data preprocessing and model evaluation stages. Its flexibility makes it an indispensable tool for communicating insights derived from machine learning analyses.

* + - **LOGISTIC REGRESSION :**

It is a foundational machine learning algorithm used for binary classification tasks. It models the probability of an instance belonging to a particular class, making it suitable for predicting outcomes like fraud or spam detection. Despite its simplicity, Logistic Regression is effective, interpretable, and widely applied in various fields due to its ability to handle linear relationships between features and target variables.

* + - **K NEAREST NEIGHBORS :**

K Nearest Neighbors (KNN) is a versatile machine learning algorithm used for classification and regression. It makes predictions based on the majority class or average of the k-nearest data points in the feature space. KNN is non-parametric, easy to implement, and doesn't require a training phase. However, it can be sensitive to the choice of k and the distance metric, impacting its performance in certain scenarios.

* + - **RANDOM FOREST:**

Random Forest is a powerful machine learning ensemble algorithm that builds multiple decision trees during training and merges their predictions for robust results. It excels in classification and regression tasks, handling complex relationships in data and mitigating overfitting. Its ability to assess feature importance and resist overfitting makes it a popular choice for various applications, including fraud detection and image classification.

* + - **SUPPORT VECTOR MACHINE (SVM) :**

Support Vector Machines (SVM) are versatile machine learning algorithms used for classification and regression tasks. SVM aims to find a hyperplane that best separates data points into distinct classes while maximizing the margin between them. It is effective in high-dimensional spaces and excels with complex decision boundaries.

SVM is widely applied in image classification, handwriting recognition, and other fields, offering robust performance in various scenarios.

* + - **XGBOOST:**

Short for eXtreme Gradient Boosting, is a powerful machine learning algorithm known for its efficiency and accuracy in both classification and regression tasks. It builds a series of decision trees iteratively, optimizing for predictive performance and handling complex relationships in data. XGBoost is widely used in various domains due to its speed, scalability, and ability to handle diverse datasets, making it a

popular choice in machine learning competitions and real-world applications.

* + - **PYTHON FLASK :**

Python Flask is a lightweight web framework widely used to deploy machine learning models as web services. With its simplicity and flexibility, Flask enables the creation of RESTful APIs, allowing seamless integration of machine learning models into web applications. Its support for handling HTTP requests and responses makes it an excellent choice for building scalable and accessible machine learning-powered applications.

DEVELOPMENT TOOLS:

* + - **Visual Studio Code :-** It is a free and open-source code editor developed by Microsoft. It's highly customizable and supports a wide range of programming languages and extensions. It provides code suggestions, autocompletion, and context- aware code hints.
    - **GIT :-** Git is a distributed version control system used for tracking changes in source code during software development. It facilitates collaboration among developers and helps manage code revisions efficiently
    - **Jupyter NoteBook :-** Jupyter Notebook is an open-source interactive web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It supports various programming languages and is widely used for data analysis, machine learning, and research.
    - **Scikit-learn :-**It is a popular machine learning library in Python. It provides a simple and efficient tool for data analysis and modeling, offering a wide range of algorithms for classification, regression, clustering, and more. Scikit- learn is widely used for its user-friendly interface, consistency, and integration with other scientific computing libraries.

#### IMPLEMENTATION

##### Data Collection and Preprocessing:

* + Gather relevant data: Obtaina comprehensive dataset of financial transactions, ensuring it includes information crucial for fraud detection.
  + Handle missing values: Implement strategies such as imputation or removal to address missing data.
  + Standardize data formats: Ensure consistency in data types and formats to facilitate subsequent analyses.

##### Exploratory Data Analysis (EDA):

* + Conduct univariate, bivariate, and multivariate analyses: Explore relationships and patterns within the data to inform feature engineering.
  + Identify outliers: Detect and address outliers that might skew the model's performance.

##### Feature Engineering:

* + Create new features: Leverage domain knowledge to engineer features that might enhance the model's predictive capabilities.
  + Transform variables: Applytransformations like log transformations to normalize skewed data.
  + Use business assumptions: Integrate insights from business assumptions to guide feature creation.

##### Data Filtering:

* + Remove unnecessary columns: Eliminate columns with no bearing on fraud detection, such as customer IDs or irrelevant timestamps.
  + Filter rows: Exclude data points that do not align with the business problem, ensuring a focused dataset.

##### Data Preparation:

* + Encode categorical variables: Convert categorical variables into numerical representations suitable for machine learning algorithms.
  + Handle imbalanced data: Employ techniques like oversampling or undersampling to address class imbalances.
  + Normalize or scale features: Enhance the model's performance by normalizing or scaling numerical features.

##### Feature Selection:

* + Use algorithms like Boruta: Apply feature selection algorithms to identify the most relevant features for model training.
  + Mitigatedimensionality: Reduce the number of features to prevent overfitting and enhance model interpretability.

##### Machine Learning Modeling:

* + Select appropriate algorithms: Choose machine learning algorithms suitable for fraud detection, such as ensemble methods or anomaly detection techniques.
  + Split data for training and testing: Allocate data for training and testing to evaluate the model's generalization capabilities.
  + Train, validate, and test the model: Assess the model's performance through rigorous

training-validation cycles and evaluate it on unseen data.

##### Hyperparameter Fine-Tuning:

* + Use grid search or random search: Systematically explore hyperparameter combinations to optimize the model's performance.
  + Cross-validate results: Validate hyperparameter choices to ensure robustness across different subsets of the data.

##### Model Evaluation and Deployment:

* + Evaluate model metrics: Assess performance metrics like precision, recall, and F1 score to understand the model's effectiveness.
  + Deploy the modelusing Flask API: Implement a Flask API to deploythe model, making it accessible for real-time fraud detection in Blocker Fraud Company's operations.

##### Continuous Improvement and Future Considerations:

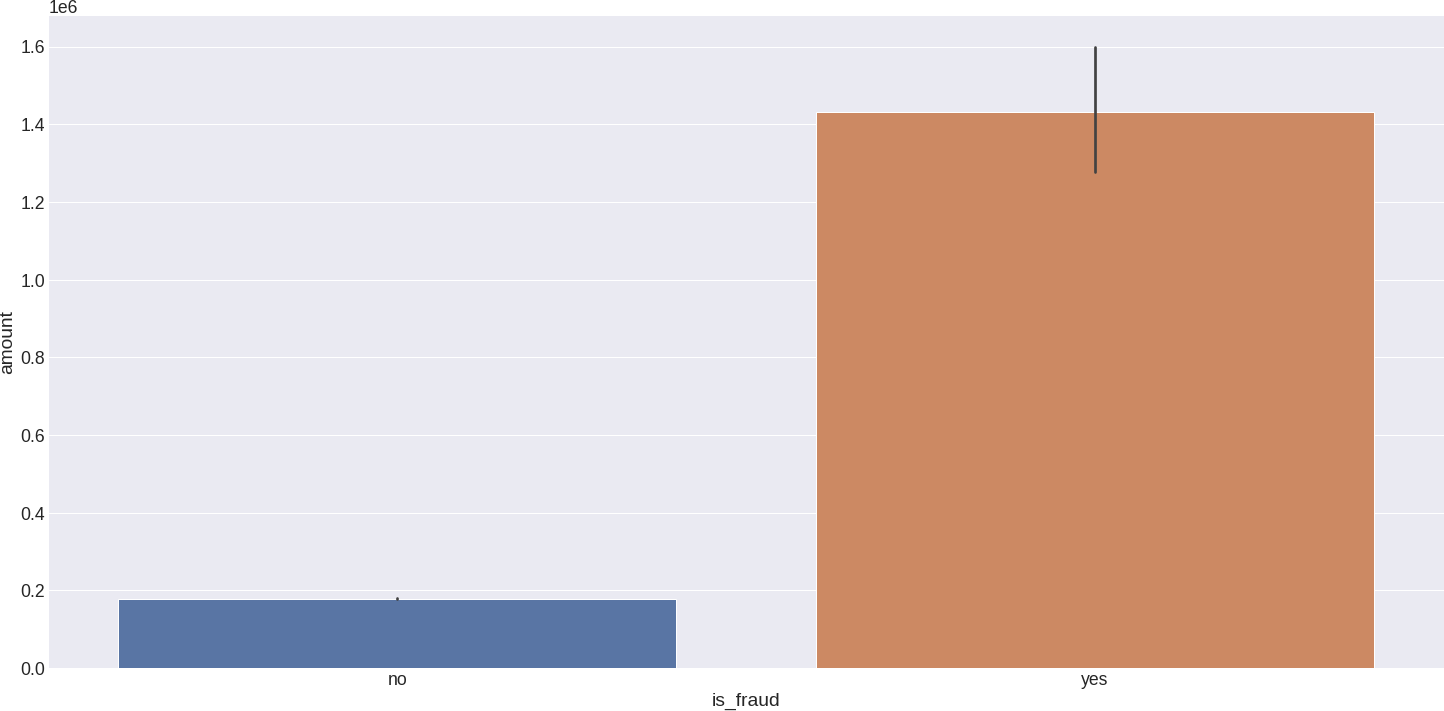
* + Monitor model performance: Regularly assess the model's performance using real-world data and update it as needed.
  + Explore emergingtechnologies: Stayabreast ofnew technologies that could enhance fraud detection capabilities.
  + Adapt to changing fraud patterns: Modify the model based on evolving fraud patterns and emerging threats.

## CHAPTER 4: RESULTS

**Discovered Insights**

### All the fraud amount is almost greater than 10.000.

The values are greater than 10.000. But it's important to note that the no-fraud values is greater than 100.000 also.

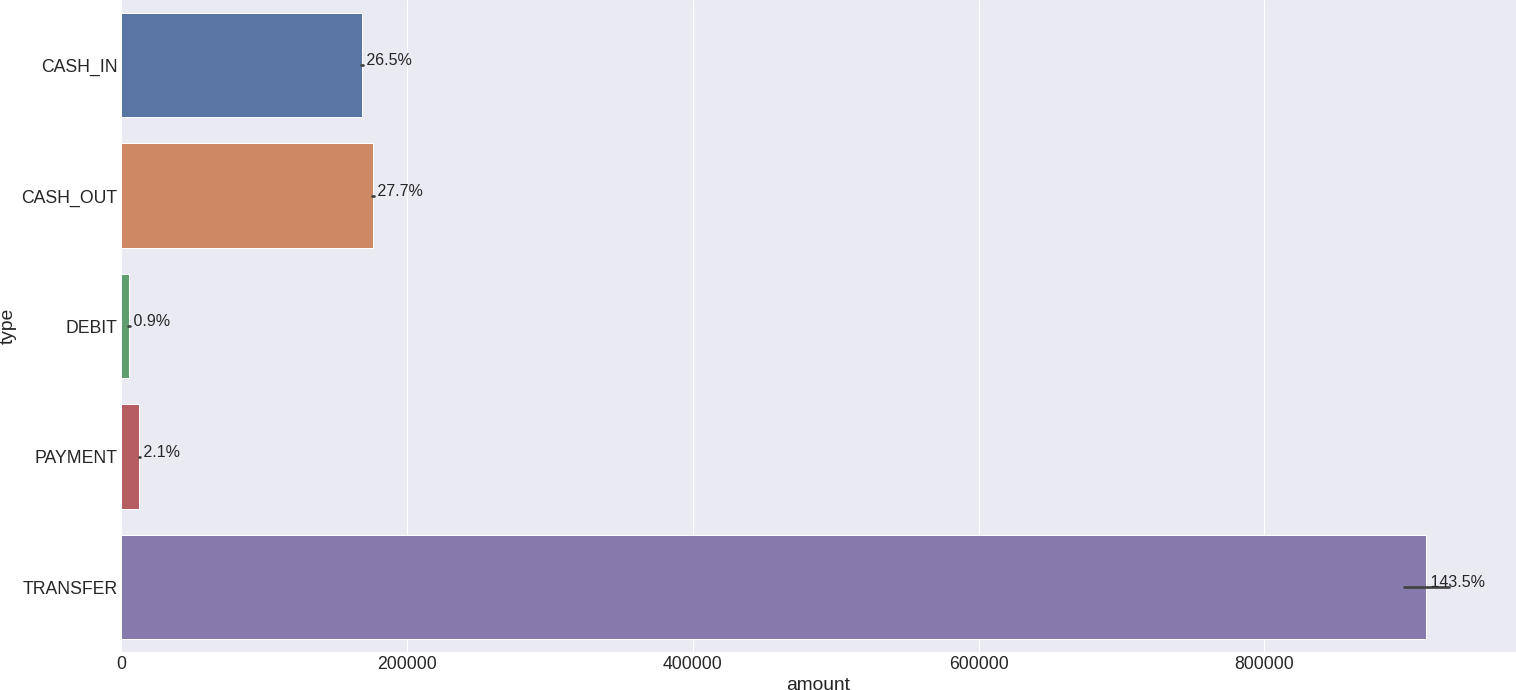


*Fig 1: Amount of frauds*

### hypothesis3The fraud transaction occurs in transfer and cash-out type. However they're almost the same value.

*Fig 2: Distribution of frauds*

### The majority transactions occurs in transfer-type, however transactions greater than 100.000 occur in cash-out and cash-in too.



*Fig 3: Distribution of Transactions*

## Cross Validation results

Here's all cross validation results of the machine learning models with their default parameters. The cross validation method is important to show the capacity of the model to learn.

### Dummy Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 4.99 +/- 0.0 | 0.0 +/- 0.0 | 0.0 +/- 0.0 | 0.0 +/- 0.0 | -0.001 +/- 0.0 |

**Logistic regression**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.565 +/- 0.009 | 1.0 +/- 0.0 | 0.129 +/- 0.017 | 0.229 +/- 0.027 | 0.228 +/- 0.027 |

### K nearest neighbors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.705 +/- 0.037 | 0.942 +/- 0.022 | 0.409 +/- 0.074 | 0.568 +/- 0.073 | 0.567 +/- 0.073 |

**Support vector Machines (SVM)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.595 +/- 0.013 | 1.0 +/- 0.0 | 0.19 +/- 0.026 | 0.319 +/- 0.0373 | 0.319 +/- 0.037 |

### Random Forest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.865 +/- 0.017 | 0.972 +/- 0.014 | 0.731 +/- 0.033 | 0.834 +/- 0.022 | 0.833 +/- 0.022 |

**XG Boost**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.88 +/- 0.016 | 0.963 +/- 0.008 | 0.761 +/- 0.033 | 0.85 +/- 0.023 | 0.85 +/- 0.023 |

**Light GBM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.701 +/- 0.089 | 0.18 +/- 0.1 | 0.407 +/- 0.175 | 0.241 +/- 0.128 | 0.239 +/- 0.129 |

## Machine Learning Performance Selection

As the results above indicate, XG Boost provides the most optimal performance and hence was chosen and was fine tuned to improve the parameters and scores further which can be observed by the table with the capacity of the model to learn below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.881 +/- 0.017 | 0.963 +/- 0.007 | 0.763 +/- 0.035 | 0.851 +/- 0.023 | 0.851 +/- 0.023 |

### Capacity to classify new data

It's possible to determine the capacity of the model to generalize using unseen data. In other words, capacity of the model to classify new data as shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Balanced Precision** | **Precision** | **Recall** | **F1** | **Kappa** |
| 0.951 | 0.944 | 0.829 | 0.883 | 0.883 |

## CHAPTER 5: CONCLUSION

In spearheading advancements in transactional fraud detection, this project champions a forward-thinking strategy, seamlessly integrating cutting-edge technologies while prioritizing an intuitive user experience. Departing from traditional methods, the platform unfolds as a transformative journey for institutions combating fraud. The incorporation of state-of-the-art technologies, prominently different machine learning algorithms to showcases the transformative capabilities of AI in conducting thorough analyses of transactional patterns and user behaviors, streamlining the fraud detection process effectively.

At its essence, it’s a modern, user-friendly interface carefully engineered for ease of use and interaction. With a lightweight and flexible backend, this interface provides a smooth user experience while positioning the system for the future. The platform’s flexibility is a testament to its commitment to adapting to the ever-changing needs of institutions and the ever-changing transactional fraud landscape.

Conclusively, this initiative will not only revolutionize transactional fraud detection, but also leave a lasting impression on strengthening financial security. Through a combination of technical ingenuity, user-centered design and ethical standards, the platform becomes a disruptive tool poised to redefine how institutions approach and protect against fraudulent activity. As financial entities continue to experience the benefits, the project and its impact are poised to reverberate, influencing the future of transactional fraud detection tools.

## CHAPTER 6: SUMMARY

The transactional fraud detection system represents a paradigm shift in fortifying financial security against fraudulent activities. Fueled by cutting-edge technologies, the system delivers a personalized and efficient platform for detecting and preventing fraudulent transactions. Different machine learning algorithms drive comprehensive analyses of transactional patterns and user behaviors, tailoring fraud detection mechanisms based on intricate insights. The frontend ensures a user-centric, dynamic, and responsive interface, garnering positive feedback during testing. The adaptable Flask backend efficiently manages data processing and communication, positioning the system for ongoing advancements. The innovative approach of personalized fraud detection distinguishes the platform, transcending conventional methods. Ethical considerations, emphasizing user privacy and legal compliance, highlight the project's commitment to integrity. Results affirm the platform's success in achieving primary objectives, providing users with an advanced, ethical, and user-friendly tool for robust transactional fraud detection.

## CHAPTER 7: FUTURE SCOPE

Future developments in transactional fraud detection have enormous potential due to the emergence of new technology and the evolution of fraud strategies. The following crucial areas determine how transactional fraud detection will develop going forward:

##### Machine Learning Advancements:

Continuous developments in machine learning algorithms will enable more sophisticated and accurate fraud detection. The integration of deep learning techniques and reinforcement learning models could significantly enhance the system's ability to adapt to new fraud patterns.

##### Real-Time Processing:

Future systems will likely prioritize real-time processing capabilities, enabling immediate detection and prevention of fraudulent transactions. This instantaneous response is crucial in staying ahead of fraudsters who continually refine their tactics.

##### Behavioral Analytics:

The incorporation of advanced behavioral analytics will become increasingly prevalent. Analyzing user behaviors and transactional patterns in-depth can offer a more nuanced understanding of normal and abnormal activities, improving the accuracy of fraud detection.

##### Integration of Blockchain:

Blockchain technology may play a role in enhancing the security of transactional processes. The decentralized and tamper-resistant nature of blockchain can provide an additional layer of protection against fraud, ensuring the integrity of transaction records.

##### Collaborative Intelligence:

Collaborative efforts among financial institutions and organizations to share data and insights on emerging fraud trends can significantly strengthen fraud detection capabilities. This collective intelligence approach fosters a united front against evolving fraud tactics.

here could make the interview preparation process more enjoyable and rewarding.

## CHAPTER 8: REFERENCES

[1] Zareapoor, M., Seeja.K.R, S. K. R., & Afshar Alam, M. (2012). Analysis on credit card fraud detection techniques: Based on certain design criteria. International Journal of Computer Applications, 52(3), 35–42. <https://doi.org/10.5120/8184-1538>

[2] Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. International Journal of Advanced Computer Science and Applications, 11(12). <https://doi.org/10.14569/ijacsa.2020.0111265>

[3] Saheed, Y. K., Hambali, M. A., Arowolo, M. O., & Olasupo, Y. A. (2020). Application of ga feature selection on Naive Bayes, random forest and SVM for credit card fraud detection. 2020 International Conference on Decision Aid Sciences and Application (DASA). <https://doi.org/10.1109/dasa51403.2020.9317228>

[4] Tanouz, D., Subramanian, R. R., Eswar, D., Reddy, G. V., Kumar, A. R., & Praneeth, C. H. V. (2021). Credit card fraud detection using machine learning. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). <https://doi.org/10.1109/iciccs51141.2021.9432308>

[5] Megdad, Mosa M. M, Abu-Naser, Samy S. & Abu-Nasser, Bassem S. (2022). Fraudulent Financial Transactions Detection Using Machine Learning. International Journal of Academic Information Systems Research (IJAISR) 6(3):30-39. <https://philpapers.org/archive/MEGFFT-2.pdf>

[6] Siddhartha Bhattacharyya, Sanjeev Jha, Kurian Tharakunnel, J. Christopher Westland, Data mining for credit card fraud: A comparative study, Decision Support Systems, Volume 50, 2011,<https://doi.org/10.1016/j.dss.2010.08.008>.

[7] Siddharth Shinde, Pravin Chavan, Dr. Mrs. K. S. Tiwari. Credit Card Fraud Detection: A Comparative Study, 2021 <https://iosrjen.org/Papers/ACCET-2021/Series-1/2.%2016-20.pdf>

[9] Leskovec, J, Rajaraman, A, & Ullman, J (2014). Data Mining In Mining of Massive Datasets (pp.1-18) Cambridge: <https://doi.org/10.1017/CBO9781139924801>

[10] Hegazy, Mohamed. (2016). Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques. Journal of Computer Science and Technology. 40. 72. <http://ecsjournal.org/Archive/Volume40/Issue3/7.pdf>

[11] Sagadevan, Saravanan & Malim, Nurul & Yee, Ong. (2018). Credit Card Fraud Detection Using Machine Learning As Data Mining Technique.

[12] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, "Credit Card Fraud Detection Using AdaBoost and Majority Voting," in IEEE Access, vol. 6, pp. 14277-14284, 2018 <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8292883>

[13] Nader Mahmoudi, Ekrem Duman, Detecting credit card fraud by Modified Fisher Discriminant Analysis, Expert Systems with Applications, Volume 42, Issue 5, 2015, Pages 2510-2516, ISSN 0957-4174, [https://doi.org/10.1016/j.eswa.2014.10.037](https://doi.org/10.1016/j.eswa.2014.10.037.)

[14] Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, Gianluca Bontempi, Learned lessons in credit card fraud detection from a practitioner perspective, Expert Systems with Applications, Volume 41, Issue 10, 2014, Pages 4915-4928, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2014.02.026.>

## CHAPTER 9: CODE

import pandas as pd import numpy as np import seaborn as sns

import matplotlib.pyplot as plt import inflection

from scipy import stats

from boruta import BorutaPy

from category\_encoders import OneHotEncoder from IPython.display import Image

from IPython.core.display import HTML

from xgboost import XGBClassifier from lightgbm import LGBMClassifier

from sklearn.svm import SVC

from sklearn.dummy import DummyClassifier

from sklearn.ensemble import RandomForestClassifier from sklearn.neighbors import KNeighborsClassifier from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import balanced\_accuracy\_score, precision\_score, classification\_report from sklearn.metrics import recall\_score, f1\_score, make\_scorer, cohen\_kappa\_score

from sklearn.preprocessing import MinMaxScaler

from sklearn.model\_selection import GridSearchCV, train\_test\_split, StratifiedKFold

*def* ml\_scores(*model\_name*, *y\_true*, *y\_pred*):

accuracy = balanced\_accuracy\_score(*y\_true*, *y\_pred*) precision = precision\_score(*y\_true*, *y\_pred*)

recall = recall\_score(*y\_true*, *y\_pred*) f1 = f1\_score(*y\_true*, *y\_pred*)

kappa = cohen\_kappa\_score(*y\_true*, *y\_pred*)

return pd.DataFrame({'Balanced Accuracy': np.round(accuracy, 3), 'Precision': np.round(precision, 3),

'Recall': np.round(recall, 3),

'F1': np.round(f1, 3),

'Kappa': np.round(kappa, 3)},

*index*=[*model\_name*])

data=pd.read\_csv('..\\data\\fraud\_0.1origbase.csv') data.describe

data.tail() data.describe() data.info()

oldColumns=data.columns.tolist() oldColumns

new\_name\_function=*lambda x*: inflection.underscore(*x*) newColumns=list(map(new\_name\_function,oldColumns)) newColumns

data.columns=newColumns data.columns

data.isna().mean()

data['is\_fraud'] = data['is\_fraud'].map({1: 'yes', 0: 'no'}) data['is\_flagged\_fraud'] = data['is\_flagged\_fraud'].map({1: 'yes', 0: 'no'})

num\_attributes = data.select\_dtypes(exclude='object') cat\_attributes = data.select\_dtypes(include='object')

num\_attributes cat\_attribute

description=num\_attributes.describe().T description

description['range']=num\_attributes.max()-num\_attributes.min()

description['variation coeffecient']=(num\_attributes.std() / num\_attributes.mean()) \*100 description['skewness']=num\_attributes.skew() description['kurtosis']=num\_attributes.kurtosis()

description

cat\_attributes.describe().T dataframe1=data.copy()

# step

dataframe1['step\_days'] = dataframe1['step'].apply(lambda i: i/24) dataframe1['step\_weeks'] = dataframe1['step'].apply(lambda i: i/(24\*7))

# difference between initial balance before the transaction and new balance after the transaction dataframe1['diff\_new\_old\_balance'] = dataframe1['newbalance\_orig'] - dataframe1['oldbalance\_org']

# difference between initial balance recipient before the transaction and new balance recipient after the transaction.

dataframe1['diff\_new\_old\_dest'] = dataframe1['newbalance\_dest'] - dataframe1['oldbalance\_dest']

# name orig and name dest

dataframe1['name\_orig'] = dataframe1['name\_orig'].apply(lambda i: i[0]) dataframe1['name\_dest'] = dataframe1['name\_dest'].apply(lambda i: i[0]) dataframe1

dataframe2=dataframe1.copy()

ax = sns.countplot(y='is\_fraud', data=dataframe2);

total = dataframe2['is\_fraud'].size for p in ax.patches:

percentage = ' {:.1f}%'.format(100 \* p.get\_width()/total) x = p.get\_x() + p.get\_width() + 0.02

y = p.get\_y() + p.get\_height()/2 ax.annotate(percentage, (x, y))

num\_attributes = dataframe2.select\_dtypes(exclude='object') columns = num\_attributes.columns.tolist()

j = 1

# Specify the number of rows and columns for the subplot grid

num\_rows = 2

num\_cols = 5

plt.figure(figsize=(15, 6)) # Create the subplot grid plt.figure(figsize=(15, 6))

for column in columns: plt.subplot(num\_rows, num\_cols, j)

sns.histplot(num\_attributes[column], kde=True) # Use sns.histplot instead of sns.displot

plt.title(column) # Add title to each subplot plt.xlabel('') # Clear x-axis label for better visualization

j += 1

plt.tight\_layout() # Adjust layout for better spacing plt.show()

cat\_attributes = dataframe2.select\_dtypes(include='object') columns = cat\_attributes.columns.tolist()

j = 1

plt.figure(figsize=(20, 10)) for column in columns:

plt.subplot(5, 2, j)

ax = sns.countplot(y=column, data=cat\_attributes)

total = cat\_attributes[column].size for p in ax.patches:

percentage = ' {:.1f}%'.format(100 \* p.get\_width()/total) x = p.get\_x() + p.get\_width() + 0.02

y = p.get\_y() + p.get\_height()/2 ax.annotate(percentage, (x, y))

j += 1

aux1=dataframe2[dataframe2['is\_fraud']=='yes'] sns.countplot(y="name\_orig",data=aux1) sns.countplot(y='name\_dest', data=aux1)

aux1

sns.barplot(y='amount', x='is\_fraud', data=dataframe2) aux1 = dataframe2[dataframe2['is\_fraud'] == 'yes']

ax = sns.countplot(y='type', data=aux1)

total = aux1['type'].size for p in ax.patches:

percentage = ' {:.1f}%'.format(100 \* p.get\_width()/total) x = p.get\_x() + p.get\_width() + 0.02

y = p.get\_y() + p.get\_height()/2 ax.annotate(percentage, (x, y))

ax = sns.countplot(y='type', hue='is\_fraud', data=dataframe2) total = dataframe2['type'].size

for p in ax.patches:

percentage = ' {:.1f}%'.format(100 \* p.get\_width()/total) x = p.get\_x() + p.get\_width() + 0.02

y = p.get\_y() + p.get\_height()/2 ax.annotate(percentage, (x, y))

ax = sns.barplot(y='type', x='amount', data=dataframe2); total = dataframe2['type'].size

for p in ax.patches:

percentage = ' {:.1f}%'.format(100 \* p.get\_width()/total) x = p.get\_x() + p.get\_width() + 0.02

y = p.get\_y() + p.get\_height()/2 ax.annotate(percentage, (x, y))

aux1 = dataframe2[dataframe2['is\_fraud'] == 'yes'] sns.regplot(x='step\_days', y='amount', data=aux1)

corr = num\_attributes.corr() mask = np.zeros\_like(corr)

mask[np.triu\_indices\_from(mask)] = True

with sns.axes\_style("white"):

ax = sns.heatmap(corr, annot=True, mask=mask, vmin=-1, center=0, vmax=1, square=True)

def calcCramerV(x, y):

cm = pd.crosstab(x, y).values n = cm.sum()

r, k = cm.shape

chi2 = stats.chi2\_contingency(cm)[0] chi2corr = max(0, chi2 - (k-1)\*(r-1)/(n-1))

kcorr = k - (k-1)\*\*2/(n-1)

rcorr = r - (r-1)\*\*2/(n-1)

try:

cramer\_v = (chi2corr / n) / (min(kcorr-1, rcorr-1))

# Check for NaN or negative values before taking the square root if np.isnan(cramer\_v) or cramer\_v < 0:

cramer\_v = 0.0 else:

cramer\_v = np.sqrt(cramer\_v)

except (ZeroDivisionError, RuntimeWarning): cramer\_v = 0.0

return cramer\_v

dict\_corr = {}

columns = cat\_attributes.columns.tolist()

for column in columns: dict\_corr[column] = {}

for column2 in columns:

dict\_corr[column][column2] = calcCramerV(cat\_attributes[column], cat\_attributes[column2]) corr = pd.DataFrame(dict\_corr)

for column2 in columns:

dict\_corr[column][column2] = calcCramerV(cat\_attributes[column], cat\_attributes[column2]) corr = pd.DataFrame(dict\_corr)

mask = np.zeros\_like(corr) mask[np.triu\_indices\_from(mask)] = True

with sns.axes\_style("white"):

ax = sns.heatmap(corr, annot=True, mask=mask, vmin=0, vmax=1, square=True) dataframe3=dataframe2.copy()

X = dataframe3.drop(columns=['is\_fraud', 'is\_flagged\_fraud', 'name\_orig', 'name\_dest', 'step\_weeks', 'step\_days'], axis=1)

y = dataframe3['is\_fraud'].map({'yes': 1, 'no': 0})

X Y

# spliting into temp and test

X\_temp, X\_test, y\_temp, y\_test = train\_test\_split(X, y, test\_size=.2, stratify=y, random\_state=42)

X\_temp X\_test

y\_temp y\_temp

# spliting into train and valid

X\_train, X\_valid, y\_train, y\_valid = train\_test\_split(X\_temp, y\_temp, test\_size=.2, stratify=y\_temp,random\_state=42)

X\_train X\_valid

y\_train y\_valid

ohe = OneHotEncoder(cols=['type'], use\_cat\_names=True) X\_train = ohe.fit\_transform(X\_train)

X\_valid = ohe.transform(X\_valid)

X\_temp = ohe.fit\_transform(X\_temp) X\_test = ohe.transform(X\_test)

num\_columns = ['amount', 'oldbalance\_org', 'newbalance\_orig', 'oldbalance\_dest', 'newbalance\_dest', 'diff\_new\_old\_balance', 'diff\_new\_old\_dest']

mm = MinMaxScaler() X\_params = X\_temp.copy()

X\_train[num\_columns] = mm.fit\_transform(X\_train[num\_columns]) X\_valid[num\_columns] = mm.transform(X\_valid[num\_columns])

X\_params[num\_columns] = mm.fit\_transform(X\_temp[num\_columns]) X\_test[num\_columns] = mm.transform(X\_test[num\_columns])

y\_temp.values.ravel()

X\_boruta = X\_params.values y\_boruta = y\_temp.values.ravel()

borutaa=BorutaPy(RandomForestClassifier(), n\_estimators='auto') # Reassigning np.int to np.int64

np.int = np.int32

# Reassigning np.float to np.float64 np.float = np.float64

# Reassigning np.bool to np.bool\_ np.bool = np.bool\_

#borutaa.fit(X\_boruta, y\_boruta) final\_columns\_selected = ['step', 'oldbalance\_org',

'newbalance\_orig', 'newbalance\_dest', 'diff\_new\_old\_balance', 'diff\_new\_old\_dest', 'type\_TRANSFER']

X\_train\_cs = X\_train[final\_columns\_selected] X\_valid\_cs = X\_valid[final\_columns\_selected]

X\_temp\_cs = X\_temp[final\_columns\_selected] X\_test\_cs = X\_test[final\_columns\_selected]

X\_params\_cs = X\_params[final\_columns\_selected]

dummy = DummyClassifier() dummy.fit(X\_train\_cs, y\_train)

y\_pred = dummy.predict(X\_valid\_cs) print(classification\_report(y\_valid, y\_pred))

def ml\_cv\_results(model\_name, model, x, y, verbose=1):

'''initial''' balanced\_accuracies = [] precisions = []

recalls = [] f1s = [] kappas = []

mm = MinMaxScaler()

x\_ = x.to\_numpy() y\_ = y.to\_numpy()

count = 0

'''cross-validation'''

skf = StratifiedKFold(n\_splits=5, shuffle=True)

for index\_train, index\_test in skf.split(x\_, y\_): ## Showing the Fold

if verbose > 0: count += 1

print('Fold K=%i' % (count))

## selecting train and test

x\_train, x\_test = x.iloc[index\_train], x.iloc[index\_test] y\_train, y\_test = y.iloc[index\_train], y.iloc[index\_test]

## applying the scale

x\_train = mm.fit\_transform(x\_train) x\_test = mm.transform(x\_test)

## training the model model.fit(x\_train, y\_train) y\_pred = model.predict(x\_test)

## saving the metrics balanced\_accuracies.append(balanced\_accuracy\_score(y\_test, y\_pred)) precisions.append(precision\_score(y\_test, y\_pred)) recalls.append(recall\_score(y\_test, y\_pred)) f1s.append(f1\_score(y\_test, y\_pred)) kappas.append(cohen\_kappa\_score(y\_test, y\_pred))

'''results'''

accuracy\_mean, accuracy\_std = np.round(np.mean(balanced\_accuracies), 3), np.round(np.std(balanced\_accuracies), 3)

precision\_mean, precision\_std = np.round(np.mean(precisions), 3), np.round(np.std(precisions), 3) recall\_mean, recall\_std = np.round(np.mean(recalls), 3), np.round(np.std(recalls), 3)

f1\_mean, f1\_std = np.round(np.mean(f1s), 3), np.round(np.std(f1s), 3)

kappa\_mean, kappa\_std = np.round(np.mean(kappas), 3), np.round(np.std(kappas), 3)

## saving the results in a dataframe

return pd.DataFrame({"Balanced Accuracy": "{} +/- {}".format(accuracy\_mean, accuracy\_std), "Precision": "{} +/- {}".format(precision\_mean, precision\_std),

"Recall": "{} +/- {}".format(recall\_mean, recall\_std), "F1": "{} +/- {}".format(f1\_mean, f1\_std),

"Kappa": "{} +/- {}".format(kappa\_mean, kappa\_std)}, index=[model\_name])

dummy\_cv = ml\_cv\_results('Dummy', DummyClassifier(), X\_temp, y\_temp) dummy\_cv

lg = LogisticRegression() lg.fit(X\_train\_cs, y\_train)

y\_pred = lg.predict(X\_valid\_cs)

lg\_results = ml\_scores('Logistic Regression', y\_valid, y\_pred) lg\_results

print(classification\_report(y\_valid, y\_pred)) lg\_cv = ml\_cv\_results('Logistic Regression',

LogisticRegression(), X\_temp\_cs, y\_temp)

lg\_cv

knn = KNeighborsClassifier() knn.fit(X\_train\_cs, y\_train)

y\_pred = knn.predict(X\_valid\_cs)

knn\_results = ml\_scores('K Nearest Neighbors', y\_valid, y\_pred) knn\_results

print(classification\_report(y\_valid, y\_pred))

knn\_cv = ml\_cv\_results('K Nearest Neighbors', KNeighborsClassifier(), X\_temp\_cs, y\_temp)

knn\_cv

svm = SVC() svm.fit(X\_train\_cs, y\_train)

y\_pred = svm.predict(X\_valid\_cs)

svm\_results = ml\_scores('SVM', y\_valid, y\_pred) svm\_results

print(classification\_report(y\_valid, y\_pred))

svm\_cv = ml\_cv\_results('SVM', SVC(), X\_temp\_cs, y\_temp) svm\_cv

rf = RandomForestClassifier(class\_weight='balanced') rf.fit(X\_train\_cs, y\_train)

y\_pred = rf.predict(X\_valid\_cs)

rf\_results = ml\_scores('Random Forest', y\_valid, y\_pred) rf\_results

print(classification\_report(y\_valid, y\_pred)) rf\_cv = ml\_cv\_results('Random Forest',

RandomForestClassifier(), X\_temp\_cs, y\_temp)

rf\_cv

xgb = XGBClassifier() xgb.fit(X\_train\_cs, y\_train)

y\_pred = xgb.predict(X\_valid\_cs)

xgb\_results = ml\_scores('XGBoost', y\_valid, y\_pred) xgb\_results

print(classification\_report(y\_valid, y\_pred))

xgb\_cv = ml\_cv\_results('XGBoost', XGBClassifier(), X\_temp\_cs, y\_temp)

xgb\_cv

lightgbm = LGBMClassifier() lightgbm.fit(X\_train\_cs, y\_train)

y\_pred = lightgbm.predict(X\_valid\_cs)

lightgbm\_results = ml\_scores('LightGBM', y\_valid, y\_pred) lightgbm\_results

print(classification\_report(y\_valid, y\_pred))

lightgbm\_cv = ml\_cv\_results('LightGDM', LGBMClassifier(), X\_temp\_cs, y\_temp)

lightgbm\_cv

modeling\_performance = pd.concat([dummy\_results, lg\_results, knn\_results, rf\_results, xgb\_results, lightgbm\_results,

svm\_results]) modeling\_performance.sort\_values(by="F1", ascending=True)

modeling\_performance\_cv = pd.concat([dummy\_cv, lg\_cv, knn\_cv, rf\_cv,

xgb\_cv, lightgbm\_cv, svm\_cv]) modeling\_performance\_cv.sort\_values(by="F1", ascending=True) f1 = make\_scorer(f1\_score)

params = {

'booster': ['gbtree', 'gblinear', 'dart'],

'eta': [0.3, 0.1, 0.01],

'scale\_pos\_weight': [1, 774, 508, 99]

}

best\_params = {'booster': 'gbtree', 'eta': 0.3, 'scale\_pos\_weight': 1} xgb\_gs = XGBClassifier(

booster=best\_params['booster'], eta=best\_params['eta'], scale\_pos\_weight=best\_params['scale\_pos\_weight']

)

xgb\_gs.fit(X\_train\_cs, y\_train) y\_pred = xgb\_gs.predict(X\_valid\_cs)

xgb\_gs\_results = ml\_scores('XGBoost GS', y\_valid, y\_pred) xgb\_gs\_results

xgb\_gs\_cv = ml\_cv\_results('XGBoost GS', xgb\_gs, X\_temp\_cs, y\_temp) xgb\_gs\_cv

final\_model = XGBClassifier( booster=best\_params['booster'], eta=best\_params['eta'], scale\_pos\_weight=best\_params['scale\_pos\_weight']

)

final\_model.fit(X\_params\_cs, y\_temp) y\_pred = final\_model.predict(X\_test\_cs)

unseen\_scores = ml\_scores('unseen', y\_test, y\_pred) unseen\_scores

## RESEARCH PAPER

Transactional Fraud Detection

Anukriti Manchanda *Maharaja Agrasen Institute of Technology,*

*Guru Gobind Singh Indraprastha University*

Delhi, India [manchandaanukriti5@gmail.com](mailto:manchandaanukriti5@gmail.com)

Anant Parashar

*Maharaja Agrasen Institute of Technology,*

*Guru Gobind Singh Indraprastha University*

Delhi, India [anantparashar005@gmail.com](mailto:anantparashar005@gmail.com)

Shubham Raturi

*Maharaja Agrasen Institute of Technology,*

*Guru Gobind Singh Indraprastha University*

Delhi, India [shubhamangwall@gmail.com](mailto:shubhamangwall@gmail.com)

Guided By:

Mr. Ajay Kumar Tiwari Assistant Professor *Maharaja Agrasen Institute of Technology,*

*Guru Gobind Singh Indraprastha*

*University*

Delhi, India [ajaytiwari@mait.ac.in](mailto:ajaytiwari@mait.ac.in)

***Abstract*— The primary objective of data analytics is to unveil concealed patterns and leverage them to facilitate well- informed decision-making across diverse scenarios. The surge in credit card fraud, propelled by technological advancements, has rendered it a vulnerable target for fraudulent activities. This poses a significant challenge in the financial services sector, incurring substantial financial losses annually, amounting to billions of dollars.**

**Developing an effective fraud detection algorithm is a complex undertaking, particularly due to the scarcity of real- world transaction datasets, attributed to confidentiality concerns and the inherent imbalance in publicly available datasets. In response to this challenge, our research addresses the issue by applying a spectrum of supervised machine learning algorithms to identify fraudulent credit card transactions, utilizing a real-world dataset.**

**Taking a step further, we harness these individual algorithms to construct a robust super classifier employing ensemble learning methods. Through our analysis, we discern the critical variables that contribute to heightened accuracy in the detection of fraudulent credit card transactions. This endeavor not only aids in enhancing the efficacy of fraud detection but also sheds light on pivotal factors influencing the success of such algorithms.**

**Moreover, our study extends beyond mere algorithmic application. We undertake a comprehensive comparative analysis, evaluating the performance of various supervised machine learning algorithms documented in the literature against the super classifier implemented in this paper.**

***Keywords— Credit Card Fraud Detection, Supervised Machine Learning, Classification, Imbalanced Dataset, Sampling Techniques, Logistic Regression, Random Forest, K- Nearest Neighbors (KNN), Dummy Classifier, Support Vector Machines, (SVM), XGBoost, LightGBM, Synthetic Dataset, Performance, Evaluation, Accuracy, Precision, F1 Score, Kappa, Recall, Feature Selection, Boruta, Hyperparameter Tuning, GridSearchCV, Confusion Matrix.***

* 1. INTODUCTION

Today, globally, data is readily accessible, with organizations of all sizes storing information characterized by high volume, variety, speed, and significance. This data originates from diverse sources such as social media interactions, user purchases [1],[2]. It is utilized for analyzing and visualizing concealed patterns in the data [3]. Initial analyses of big data primarily focused on data volume, encompassing general public databases [4], biometrics, and financial analyses [5], [6].

In the realm of fraud, the realm of transactions proves tobe a convenient and inviting target due to the substantial monetary gain achievable within a brief timeframe [7]. Perpetrators of transactional fraud aim to pilfer sensitive information, including transactional numbers, bank account details, and social security numbers [8], [6]. The endeavor to make each fraudulent transaction appear legitimate presents a formidable challenge in fraud detection [9]. The escalation in transactional dataset transactions indicates that around 70% of individuals in the US are susceptible to falling into the snares set by these fraudsters [10].

Transactional datasets often exhibit a significant imbalance, containing a higher volume of legitimate transactions compared to fraudulent ones [11]. Consequently, predictions may yield a notably high accuracy score without effectively identifying fraudulent transactions. Addressing this issue involves balancing class distribution through techniques like sampling minority classes [12]. In such sampling methods, training examples from the minority class are augmented in proportion to the majority class, enhancing the algorithm's capability to make accurate predictions [13].

This study employs seven machine learning models, evaluating their Accuracy, Recall, Precision, Kappa, and F1-Score. All machine learning algorithms undergo assessment using a synthetic dataset generated to mirror real-world scenarios and discern between fraudulent and non-fraudulent transactions. The primary objective of this study is to employ supervised learning methods on authentic datasets [14]

* 1. RELATED STUDY

Related study Utilizing logistic regression and artificial neural networks, the system identifies fraudulent and legitimate transactions based on their transaction scores. However, the overall performance of all machine learning models is adversely affected by the skewness present in the training dataset [15].

To address the issue of an unbalanced dataset, two distinct methods have been employed: intrinsic features and network-based features [16]. Intrinsic features involve a comparison of a customer's past transactions to identify any discernible patterns. On the other hand, network-based features leverage the connections among credit card holders and merchants, assigning a time-dependent suspiciousness score to each network object. These approaches yield a remarkably high accuracy score in Random Forest, achieving a mere 1% false positive rate, thereby creating an almost flawless model for detecting fraudulent transactions [11].

Comparative analyses were conducted across different modeling and algorithmic techniques using a real dataset, revealing that certain algorithms underperformed due to the dataset's unbalanced nature [11]. Addressing unbalanced datasets from both non-stream credit card and data streams, three distinct methods were employed: static, update, and DataStream. Additionally, two undersampling methods, namely SMOTE and Easy Ensemble, were applied to balance the dataset [17]. Notably, in Random Forest (RF) and Support Vector Machine (SVM), a decrease in the Area Under the Curve (AUC) was observed alongside an increase in F-measure [18].

The neural network architecture, employed in an unsupervised manner using real-time transaction entries [4], involves the utilization of a self-organizing map. Through optical classification, this map resolves issues associated with each specific group [19], achieving a 95% fraud detection rate with a ROC curve and without triggering false alarms.

Data Mining reports the development and implementation of a fraud detection system in a large e-tail merchant [20]. Using a cost-based performance, the algorithm is trained to obtain business outcomes, albeit requiring a longer training time [21]. A bank seller decision support system is utilized for banking fraud analysis and investigation. This system automatically detects fraud, assigns ranks, and comprehends user spending habits based on past transactions, employing mathematical and statistical techniques [22].

* 1. OUR APPROACH AND METHODOLOGY

1. Data Collection and Preprocessing:

* Gather relevant data: Obtain a comprehensive dataset of financial transactions, ensuring it includes information crucial for fraud detection.
* Handle missing values: Implement strategies such as imputation or removal to address missing data.
* Standardize data formats: Ensure consistency in data types and formats to facilitate subsequent analyses.

1. *Exploratory Data Analysis (EDA):*

* Conduct univariate, bivariate, and multivariate analyses: Explore relationships and patterns within the data to inform feature engineering.
* Identify outliers: Detect and address outliers that might skew the model's performance.

1. *Feature Engineering:*

* Create new features: Leverage domain knowledge to engineer features that might enhance the model's predictive capabilities.
* Transform variables: Apply transformations like log transformations to normalize skewed data.
* Use business assumptions: Integrate insights from business assumptions to guide feature creation.

1. *Data Filtering:*

* Remove unnecessary columns: Eliminate columns with no bearing on fraud detection, such as customer IDs or irrelevant timestamps.
* Filter rows: Exclude data points that do not align with the business problem, ensuring a focused dataset.

1. *Data Preparation:*

* Encode categorical variables: Convert categorical variables into numerical representations suitable for machine learning algorithms.
* Handle imbalanced data: Employ techniques like oversampling or undersampling to address class imbalances.
* Normalize or scale features: Enhance the model's performance by normalizing or scaling numerical features.

1. *Feature Selection:*

* Use algorithms like Boruta: Apply feature selection algorithms to identify the most relevant features for model training.
* Mitigate dimensionality: Reduce the number of features to prevent overfitting and enhance model interpretability.

1. *Machine Learning Modeling:*

* Select appropriate algorithms: Choose machine learning algorithms suitable for fraud detection, such as ensemble methods or anomaly detection techniques.
* Split data for training and testing: Allocate data for training and testing to evaluate the model's generalization capabilities.
* Train, validate, and test the model: Assess the model's performance through rigorous training-validation cycles and evaluate it on unseen data.

1. *Hyperparameter Fine-Tuning:*

* Use GridSearchCV: Systematically explore hyperparameter combinations to optimize the model's performance.
* Cross-validate results: Validate hyperparameter choices to ensure robustness across different subsets of the data.

1. *Model Evaluation:*

* Evaluate model metrics: Assess performance metrics like precision, recall, and F1 score to understand the model's effectiveness.

1. *Continuous Improvement and Future Considerations:*

* Monitor model performance: Regularly assess the model's performance using real-world data and update it as needed.
* Explore emerging technologies: Stay abreast of new technologies that could enhance fraud detection capabilities.
* Adapt to changing fraud patterns: Modify the model based on evolving fraud patterns and emerging threats.
  1. RESULTS

Our research delved into transactional fraud detection, systematically evaluating machine learning models for their effectiveness in handling imbalanced datasets prevalent in fraud scenarios. In the initial cross-validation phase, XGBoost emerged as a standout performer, showcasing promising metrics in balanced accuracy, precision, recall, F1 score, and kappa.

Following this success, we fine-tuned the XGBoost model, resulting in exceptional performance metrics - an 88.1% balanced accuracy, 96.3% precision, 76.3% recall, and an 85.1% F1 score and kappa. This demonstrated the model's heightened ability to accurately identify and classify fraudulent transactions, suggesting its practical viability.

In validation on a test dataset, the final tuned XGBoost model excelled with a 91.5% balanced accuracy, 94.4% precision, 82.9% recall, and an 88.3% F1 score and kappa. These results underscore the model's consistent and resilient performance, highlighting its potential for real- world applications in transactional fraud detection.

Our study emphasizes the critical role of meticulous model selection and parameter tuning in developing effective fraud detection systems. XGBoost, with its adaptable ensemble learning approach, stands out as a potent tool, particularly adept at addressing challenges posed by imbalanced datasets in transactional fraud detection.

1. *Discovered Insights*
   1. *All the fraud amount is almost greater than 10.000. The values are greater than 10.000.*

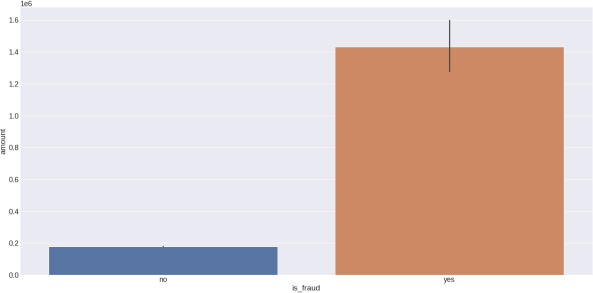


Fig. 1. Amount of frauds

* 1. *The fraud transaction occurs in transfer and cash- out type. However they're almost the same value.*

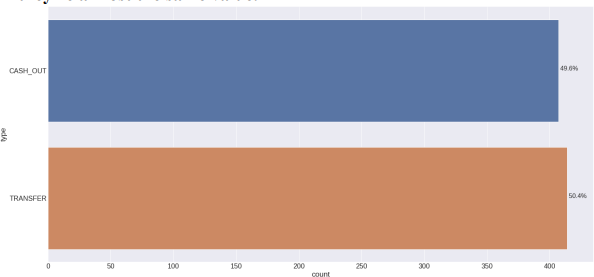
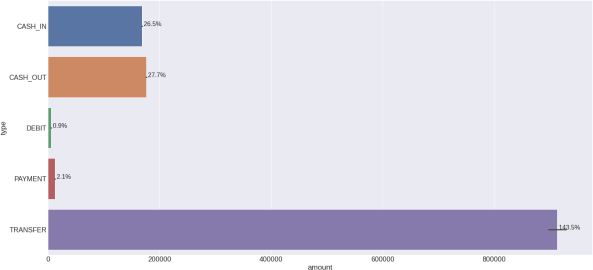


Fig. 2. Distribution Of Frauds

* 1. *The majority transactions occurs in transfer-type, however transactions greater than 100.000 occur in cash- out and cash-in too.*

Fig. 3. Distribution Of Transactions



1. *Cross Validation results*

TABLE I. DUMMY MODEL

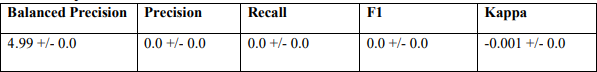


TABLE II. LOGISTIC REGRESSION

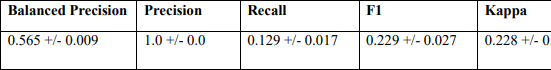


TABLE III. K-NEAREST NEIGHBORS

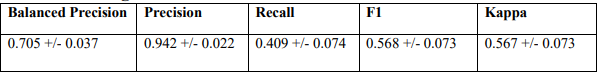


TABLE IV. SUPPORT VECTOR MACHINE (SVM)

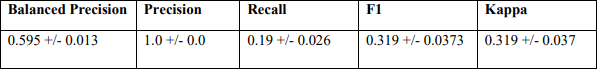


TABLE V. RANDOM FOREST

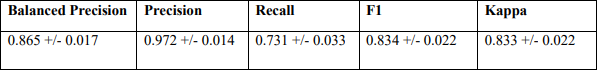


TABLE VI. XG BOOST

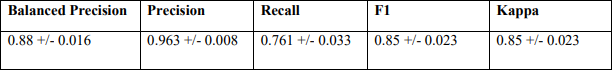


TABLE VII. LIGHT GBM

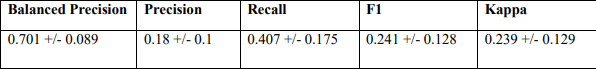
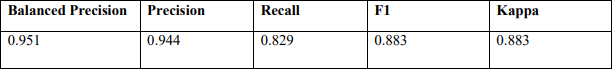


TABLE VIII. FINAL MODEL-XG BOOST WITH HYPERPARAMETER TUNING



* 1. CONCLUSIONS

In spearheading advancements in transactional fraud detection, this project champions a forward-thinking strategy, seamlessly integrating cutting-edge technologies while prioritizing an intuitive user experience. Departing from traditional methods, the platform unfolds as a transformative journey for institutions combating fraud. The incorporation of state-of-the-art technologies, prominently different machine learning algorithms to showcases the transformative capabilities of AI in conducting thorough analyses of transactional patterns and user behaviors, streamlining the fraud detection process effectively. At its essence, it’s a modern, user-friendly interface carefully engineered for ease of use and interaction. With a lightweight and flexible backend, this interface provides a smooth user experience while positioning the system for the future. The platform’s flexibility is a testament to its commitment to adapting to the ever-changing needs of institutions and the ever- changing transactional fraud landscape. Conclusively, this initiative will not only revolutionize transactional fraud

detection, but also leave a lasting impression on strengthening financial security. Through a combination of technical ingenuity, user-centered design and ethical standards, the platform becomes a disruptive tool poised to redefine how institutions approach and protect against fraudulent activity. As financial entities continue to experience the benefits, the project and its impact are poised to reverberate, influencing the future of transactional fraud detection tools.

* 1. REFERENCES

1. S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” Decis. Support Syst., vol. 50, no. 3, pp. 602–613,

2011.

1. K. Chaudhary, J. Yadav, and B. Mallick, “A review of Fraud Detection Techniques: Credit Card,” Int. J. Comput. Appl., vol. 45, no. 1, pp. 975–8887, 2012.
2. “Mining of Massive Datasets Second Edition.”
3. F. N. Ogwueleka, “Data Mining Application in Credit Card Fraud Detection System,” vol. 6, no. 3, pp. 311–322, 2011.
4. H. Nordberg, K. Bhatia, K. Wang, and Z.Wang, “BioPig: a Hadoop-based analytic toolkit for large-scale sequence data,” Bioinformatics, vol. 29, no. 23, pp. 3014– 3019, Dec. 2013.
5. M. Hegazy, A. Madian, and M. Ragaie, “Enhanced Fraud Miner: Credit Card Fraud Detection using Clustering Data Mining Techniques,” Egypt. Comput. Sci., no. 03, pp. 72–81, 2016.
6. M. Zareapoor and P. Shamsolmoali, “Application of credit card fraud detection:Based on bagging ensemble classifier,” Procedia Comput. Sci., vol. 48, no. C, pp. 679– 686, 2015.
7. O. S. Yee, S. Sagadevan, N. Hashimah, and A. Hassain, “Credit Card Fraud Detection Using Machine Learning As Data Mining Technique,” vol. 10, no. 1, pp. 23–27.
8. K. Randhawa, C. K. Loo, M. Seera, C. P. Lim, and A.

K. Nandi, “Credit Card Fraud Detection Using AdaBoost and Majority Voting,” IEEE Access, vol. 6, pp. 14277– 14284, 2018.

1. N. Mahmoudi and E. Duman, “Detecting credit card fraud by Modified Fisher Discriminant Analysis,” Expert Syst. Appl., vol. 42, no. 5, pp. 2510–2516, 2015.
2. A. Dal Pozzolo, O. Caelen, Y. A. Le Borgne, S. Waterschoot, and G. Bontempi, “Learned lessons in credit card fraud detection from a practitioner perspective,” Expert Syst. Appl., vol. 41, no. 10, pp. 4915–4928, 2014.
3. M. A. Scholar, M. Ali, and P. Fellow, “Investigating the Performance of Smote for Class Imbalanced Learning: A Case Study of Credit Scoring Datasets,” vol. 13, no. 33, pp. 340–353, 2017.
4. H. He, W. Zhang, and S. Zhang, “A novel ensemble method for credit scoring: Adaption of different imbalance ratios,” Expert Syst. Appl., vol. 98, pp. 105–117,

May 2018.

1. A. Dal Pozzolo, O. Caelen, R. A. Johnson, and G. Bontempi, “Calibrating Probability with Undersampling for Unbalanced Classification.”
2. Y. Sahin and E. Duman, “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines,” Int. Multiconference Eng. Comput. Sci., vol. I, pp. 442–447, 2011.
3. V. Van Vlasselaer et al., “APATE: A novel approach for automated credit card transaction fraud detection using networkbased extensions,” Decis. Support Syst., vol. 75, pp. 38–48, 2015.
4. C. Phua, D. Alahakoon, and V. Lee, “Minority report in fraud detection,” ACM SIGKDD Explor. Newsl., vol. 6, no. 1, p. 50, 2004.
5. E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature,” Decis. Support Syst., vol. 50, no. 3, pp. 559–569, 2011.
6. C. C. Lin, A. A. Chiu, S. Y. Huang, and D.

C. Yen, “Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts’ judgments,” Knowledge-Based Syst., vol. 89, pp. 459–470, 2015.

1. N. Carneiro, G. Figueira, and M. Costa, “A data mining based system for credit-card fraud detection in e- tail,” Decis. Support Syst., vol. 95, pp. 91–101, 2017.

## PUBLICATION PROOF

