

***Fraud Detection in Financial Transactions: A Machine Learning Approach Using IEEE-CIS Dataset***

Shazib Armaan Patras

B01750852

Thesis for the partial fulfilment

of the requirements for the Masters Degree

in MSc Information Technology

University of the West of Scotland

School of Computing, Engineering and Physical Sciences

21-04-2025

A close up of a logo

Description automatically generated

**DECLARATION OF ORIGINALITY**

I declare that this is an original study based on my own

work and that I have not submitted it for any other

course or degree.

Signature\_\_\_\_\_\_\_\_\_\_SHAZIB\_\_\_\_\_\_\_\_\_\_\_\_\_

****

Library Form to Accompany MSc Project

To be completed in full

|  |
| --- |
| **Surname:** **Patras** |
| **First Name: Shazib Armaan Initials: SP** |
| **Banner No: B01750852** |
| **Course Code: COMP11117** |
| **Course Title: MSc IT with Data Analytics** |
| **Project Supervisor: Martins Olaleye** |
| **Project Title: *Fraud Detection in Financial Transactions: A Machine Learning Approach Using IEEE-CIS Dataset*** |
| **Session: 2024/2025 Date of Submission: 21/04/2025** |
| **Signature: SHAZIB** |

**Abstract**

*An important aspect of the current financial world is that online transaction fraud costs billions of dollars every year. This project focuses on investigating machine learning for fraud detection on the IEEE-CIS dataset (590,540 transactions; 3.5% fraud). By combining statistical analysis with modern machine-learning techniques, we will see whether the amounts involved in transactions are correlated with fraud and build a predictive model that improves detection. First, this involved preprocessing the data (joining identity and transaction data, managing more than four hundred features with missing values and categorical encoding). The exploratory data analysis revealed interesting patterns: fraudulent transactions have a slightly higher median amount, as well as differing temporal behaviors (fraud peaks earlier in the morning) from the genuine ones. We conduct tests of our hypotheses (Mann-Whitney U, Welch’s t-test), which confirm that there exists a statistically significant difference between transaction amounts in fraud and non-fraud cases (p < 0.001), though there is a large overlap between their distributions themselves. Next, we build several machines learning models, including gradient-boosted tree methods (LightGBM, CatBoost, XGBoost), as well as a neural network, with stratified training and evaluation of ROC AUC and precision-recall metrics. Thus, our best model (XGBoost ensemble) achieved AUC 0.95 and 95% precision on the fraud class, outperforming the baseline rules. Weights of evidence for explanations, through Shapley values (SHAP), suggest transaction amount, card and address matches, and device info are important. Practical aspects of deployment are discussed, including class imbalance treatment, real-time inference speed, and the inherent practical trade-off between false positives and negatives. Conclusive recommendations are made about the integration of ML-based fraud detection in financial systems, pursued with ethical considerations and future research avenues on adaptive and hybrid techniques.*

Contents

[1. Introduction 7](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945836)

[1.1 Research Focus and Dataset 7](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945837)

[1.2. Research Questions 8](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945838)

[1.3. Methodology 8](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945839)

[1.4. Significance and Expected Contributions 9](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945840)

[1.5. Report on Structure 9](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945841)

[2. Literature Review 11](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945842)

[2.1. The Evolution of Fraud Detection Techniques 11](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945843)

[2.2 Machine Learning Techniques and New Developments 12](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945844)

[2.3 Statistical and Hybrid Approaches 14](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945845)

[2.4 Gaps and Current Research Trends 15](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945846)

[3. Methodology 17](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945847)

[3.1 Data Collection and Integration 17](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945848)

[3.2 Data Preprocessing and Cleaning 18](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945849)

[3.2.1 Memory optimization: 18](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945850)

[3.2.2 Missing Values: 18](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945851)

[3.2.3 Feature encoding: 19](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945852)

[3.2.4 Feature scaling and transformation: 20](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945853)

[3.3 Exploratory Data Analysis (EDA) 20](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945854)

[3.4 Machine Learning Modeling Approach 25](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945855)

[3.5 Model Interpretation (SHAP) and Feature Importance 29](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945856)

[3.6 Statistical Hypothesis Testing 33](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945857)

[4. Results 35](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945858)

[4.1 Exploratory Analysis Results 35](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945859)

[4.2 Model Performance and Benchmarking 35](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945860)

[4.3 Feature Importance and Model Insights 36](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945861)

[4.4 Challenges and Limitations 38](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945862)

[4.5 Implications for Fraud Detection 38](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945863)

[5. Discussion 40](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945864)

[5.1 Transaction Amount of Fraud 40](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945865)

[5.2 Model Performance and Feature Engineering Insights 40](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945866)

[6. Conclusion 47](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945867)

[7. References 50](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945868)

[8. Appendices 53](file:///C:\Users\shazi\Desktop\EXAM\Grad\UWS\Thesis\B01750852_Shazib_Armaan_Patras_thesis.docx#_Toc195945869)

# Introduction

Financial transaction fraud is one of the most damaging threats to the digital economy today. It creates enormous economic fallouts internationally. In fact, total payment card fraud losses across the world reached an alarming $34 billion last year alone. Consequently, this creates a huge call for more effective detection systems. Traditional fraud prevention approaches applying to banks and payment processors used rules measures defined by experts to identify purchases uncharacteristic of the customer concerned or made from a foreign location and complement it with manual review. Although these methods have been standard in the industry for several decades, they, too, have their limitations, especially in adapting to the real-time changing patterns of sophisticated fraud schemes. In addition, they tend to generate excessive false positives that hinder smooth customer experience with legitimate transactions appearing as suspicious ones.

Traditional methods have reached their limits, and this catalyzed the change in basic assumptions toward machine learning (ML) based approaches for fraud detection. ML has produced a tempting option to automatically learn the complex patterns of fraudulent behavior from historical transactional data without explicit programming. The operational organizations that have installed the ML techniques in their fraud detection systems have experienced a much better detection rate compared to rule-based approaches. However, it is important to note that these technologies are fraught with challenges. The two most prominent concerns are the interpretation of model decisions, which is crucial for regulatory compliance and stakeholder trust, and the inbuilt class imbalance in fraud datasets, with legitimate transactions typically outnumbering fraudulent ones at a ratio of as high as 20:1 (these fraud cases comprise less than 5% of all transactions).

## 1.1 Research Focus and Dataset

This work investigates the domain of fraud detection in financial transactions using the IEEE-CIS Fraud Detection Dataset, released as part of the Kaggle competition in 2019. The dataset contains 590,000 online transactions with binary labels indicating fraud status and was contributed by Vesta Corporation (a leading payment service provider). Unique features comprising almost 431 device information, product characteristics, and identity attributes were asked for anonymity and competitive reasons. An incredibly challenging aspect of this dataset is the class imbalance, about 3.5% of transactions are considered as fraudulent, closely matching real-world conditions in the financial industry.

## 1.2. Research Questions

In our study, we will attempt to answer two fundamental research questions. The first is Statistical Insight, which asks: Is there a statistically significant relationship between transaction amount and the likelihood of fraud? This question addresses a commonly held assumption in the domain of financial security regarding the proposition that higher-value transactions carry the greater risk of fraud offsetting its profit: this is an assumption that may motivate current schemes, but we shall evaluate its validity rigorously through empirical investigations. The second question is whether recent machine-learning techniques in general and gradient boosting algorithms and neural networks in particular can detect fraudulent transactions on highly imbalanced datasets. We want to build models to balance fraud detection maximization (high recall) and minimizing false alerts (high precision) and to comparatively analyze their different merits over a variety of performance aspects.

## 1.3. Methodology

A dual methodological scheme which addresses the questions in an all-inclusive manner will be employed.

First, we perform Exploratory Data Analysis of transaction amounts and other potentially discriminating factors in depth supported by statistical hypothesis testing. Distinguishing patterns responsible for fraud from legitimate transactions will be profiled, thus providing domain-relevant insights useful for model design and strategy development.

On the second front, we build and examine several machine-learning models dedicated to fraud detection. For the modeling study, we use advanced ensemble tree-based methods such as LightGBM, CatBoost, and XGBoost considering their recent excellent performance on tabular data. We further implement a neural network framework to perform comparisons. During the modeling study, all the model performances were reported through metrics specifically developed for imbalanced cases such as ROC-AUC and precision-recall curves, as accuracy could be misleading due to extreme class imbalance. We utilize Shapley additive explanations (SHAP) for model interpretability to enhance the transparency and trustability of the models so that the reasons behind predictions can be clearly communicated to non-technical stakeholders such as business executives, compliance officers, and customers.

## Significance and Expected Contributions

There are important implications for improving fraud detection on more than just a financial level, including operations and society. Direct consequences of improved fraud detection include reductions in the money that is lost directly from fraud, reductions in operating costs of investigations, and increased customer trust and experience. Not-so-great detection systems have other great negative consequences: false negatives (fraud cases that are not caught) lead to unrecovered financial losses, and false positives (also known as legitimate transactions incorrectly flagged as fraudulent) frustrate customers, ruin brand reputation, and incur unnecessary dollar operational expense for review.

The contributions that our study will make are threefold.

1. Our rigorous statistical analyses will testify to the quantification of the association between transaction value and potential fraud risk; thus, bringing evidence-based insights into managing risks.

2. Implement and evaluate the most efficient fraud detection models developed for real-world conditions of class imbalance and feature complexity.

3. Finally, provide operational application direction for such models with emphasis and distinction trade-off between detection sensitivity, false positive rates, and computational efficiency, as well as model interpretability. These contributions fall into the long-range objective of improving the capabilities of fraud prevention while balancing careful risk management imperatives with customer experience considerations.

## Report on Structure

The remainder of this report is structured as follows:

The Literature Review surveys recent studies (ranging from 2018 to 2024) on fraud detection methodologies, focusing specifically on machine learning-based approaches, statistical methods, and hybrid systems. This section puts our work into context about the existing academic and industrial landscape, while also highlighting the specific gaps that our project is filling.

Methodology presents detailed information related to the characteristics of the datasets, preprocessing, exploratory data analysis (EDA), modeling techniques, and validation strategies. It also details our statistical testing framework and SHAP analysis processes.

Results present our empirical findings through EDA insights, comparative metrics of model performance with visualizations such as ROC and precision-recall curves, confusion matrices, feature importance explanations, and hypothesis test results.

To interpret these results in a wider perspective of financial fraud detection, as well as by discussing practical deployment considerations (including model inference speed, update mechanisms, and false positive management strategies), ethical issues in automated decision systems in financial services are also included in the Discussion.

Finally, the Conclusion gives a summary of the key points and outlines good possibilities for future work, such as investigating graph neural networks to capture transaction relations, or federated learning to increase privacy.

# Literature Review

The efforts related to fraud detection have increased significantly over decades. It originated from manual inspections to simple rule-based systems to higher, advanced machine learning hybrid techniques. In summary, this article addressed literature from 2018 to 2024, covering three aspects of fraud detection systems: (i) current machine learning approaches of fraud detection, (ii) statistical and anomaly detection techniques, and (iii) hybrid and emerging models with a mixture of techniques. The article discusses recent progress made and outstanding challenges identified in literature.

## 2.1. The Evolution of Fraud Detection Techniques

Most of the early fraud detection systems depended on statistical methods and expert rules by which the domain experts specified thresholds or red-flag conditions, such as transactions above a given amount or that occur with a specific frequency. Bolton and Hand (2002) have made an important and foundational review of statistical fraud detection techniques in which statistical techniques are categorized into unsupervised techniques (peer group analysis, outlier detection) and supervised classification methods that were available at that time. As the data volumes increased, data mining and machine learning approaches gained popularity. This trend has been chronicled by surveys such as Abdallah et al. (2016) and West and Bhattacharya (2016), who reported the trend from turning away traditional statistical models in favor of algorithms such as neural networks, decision trees, and ensemble methods. Challenges were identified in these surveys- the challenges that are still existent, such as the handling of imbalanced data, non-stationary patterns, for instance, adaptation in the strategies, and interpretability in the auto decision making.

For the past five years, the research thrust concentrated remarkably on machine learning (ML) ways for fraud detection because of increases in computing power combined with rich datasets. Logistical regressions, support vector machines (SVMs), and random forests rank among the classical ML algorithms applied to credit cards and online transaction fraud detection. For instance, Awoyemi et al. (2017) proved tree-based models optimal with proper handling of class imbalance when comparing logistic regression, naïve Bayes, and decision tree models towards credit card fraud detection. Gradient boosting frameworks such as XGBoost, LightGBM, and CatBoost are promising for drawing more popularity for providing high accuracy while dealing with large-scaled, high-dimensional data. XGBoost, a gradient boosting method initiated by Chen and Guestrin (2016), was a great contribution to the field and has been used in many fraud detection studies and competitions for its efficiency and accuracy. LightGBM (Ke et al., 2017), CatBoost (Prokhorenkova et al., 2018) improved upon gradient boosting further; CatBoost manages categorical features natively-an edge where transaction data have many categorical fields, such as card type, email domain, etc. Gradient boosting has been practically accepted as the best performing method for detecting credit card fraud, often beating much simpler models.

Class imbalance is, however, a crucial problem in the conventional ML applications because fraud itself is rare; hence models may achieve high overall accuracy by simply predicting all transactions such as falling under the majority or non-fraud class. This has led to many researchers resorting to resampling and cost-sensitive learning approaches to handle issues regarding class imbalance. Dal Pozzolo and others (2015) share some lessons from practitioner-oriented applications with recommendations for under sampling the majority class and carefully calibrating decision thresholds to optimize the recall rate at an acceptable threshold for false positives.

Makki et al. (2019) compared under sampling, oversampling (SMOTE), and ensemble methods in their experimental study of imbalanced classification methods for credit card fraud detection. The authors found out that their performance improved further by combining SMOTE (Synthetic Minority Over-sampling Technique) with ensemble classifiers, as also seen in other applications that synthetic oversampling could be extremely helpful in boosting recall. Another alternative is to assign a higher misclassification cost to fraud instances: whereas Bahnsen et al. (2016), they integrated cost-sensitive learning by adjusting decision thresholds according to fraud loss metrics.

## 2.2 Machine Learning Techniques and New Developments

Today we apply more deep learning models to fraud detection purposes since about 2018 as an additional measure increased their attractiveness in modelling complex, nonlinear patterns. They utilized deep autoencoders and restricted Boltzmann machines in an unsupervised scheme to detect credit card fraud, achieving great improvements in accuracy. Their research is an example of the truly unsupervised machine learning techniques, such as use of neural networks in anomaly detection, to identify transactions that differ from the learned "normal" ones as fraudulent. Another challenge is that purely unsupervised approaches often fail to tell the difference between fraud and legitimate but strange behavior. Accordingly, researchers have developed mixed deep learning approaches to solving the problem. Jurgovsky et al. (2018) considered credit card transactions as sequential by modeling them with recurrent neural networks (LSTMs) for transaction sequences, thus framing the problem in detection as that of a sequence classification. They reported that novel sequence-based deep models could trap fraud by considering temporal spending patterns (e.g., rapidly made expenditures) missed by single-transaction models.

The use of Graph-based machine learning for fraud detection is another new horizon. Fraud usually occurs in rings or networks, say, with multiple cards linked to a single device or IP address) such analysis can reveal suspicious nodes. Van Vlasselaer et al. (2015) used an approach to detect credit card fraud using network characteristics and link analysis that established the utility of network-based features in improving performance. In the follow-up works on fraud detection utilizing GNNs, Wang et al. (2019) and Cheng et al. (2020) treat transactions as edges in a graph, learning embeddings for entities to reveal illegal matters. The results show that complexity is not an enemy of these GNNs, as they are. In the papers by Tang et al. published in 2024, we find an attempt to couple GNNs with federated learning to allow several institutions to jointly train a fraud model while protecting sensitive data. This is a very promising endeavor toward industry deployment with a strong emphasis on data privacy.

Ensemble learning is another method popular in recent literature. Many of the best solutions, including those of the IEEE-CIS competition winners, use ensembles of diverse models (e.g., blending neural networks with gradient boosters) to capture various aspects of the data. Fiore et al. (2019) proposed the use of GANs to generate synthetic fraudulent transactions to augment the training data. In finding the real examples of fraud generated by the GAN model, they have improved the classifier's ability to detect real fraud. Thus, GANs are demonstrated to be a valuable tool for data augmentation and anomaly detection in the field of fraud. These models with GAN augmentation and other meta-learning techniques (stacking multiple classifiers, for instance) constitute some of the ways in which researchers leverage multiple methods to increase fraud detection performance.

Nonetheless, according to various recent studies, challenges still exist, such as interpretability and real-time performance of the models. In their systematic literature review, Hafez et al. (2025) states that the mainstream deep learning models (CNNs, RNNs, transformers), notwithstanding their firepower, are usually referred to as "black boxes", calling for interpretable AI in fraud detection. Thus, methods like SHAP (2017) and LIME find traction alongside complex models in explaining their predictions to stakeholders or satisfying requirements set by regulators. Another challenge noted was concept drift: fraud patterns evolve as fraudsters adapt to detection, thus requiring models to be monitored and updated continuously. Research by Rotman and Shapira (2020) (as cited in newer reviews) has delved into adaptive learning and drift detection algorithms to keep fraud models updated. In summary, in the ML-centered literature about fraud detection from 2018 to 2024, we see a clear trend: more sophisticated model research (deep learning, GNN) and data-centric techniques (oversampling, synthetic data, cost-sensitive loss) drive greater detection performance.

## 2.3 Statistical and Hybrid Approaches

While statistical analysis has always been and still is of utmost interest to fraud analysts and is often combined with machine learning, most researchers have emphasized what intrinsic factors lead to a suspicion of transactions. One such variable of interest is the number of transactions. Interestingly, few studies focus solely on transaction amount as a predictor; it is usually one of many features in the machine learning model. This study thus attempts to bridge the gap regarding the association between amount and fraud. Bhattacharyya et al. (2011) remarked that the addition of transaction amounts improved their models, but the variable alone had scant power in predicting fraud. Likewise, Whitrow et al. (2009) found that recent spending behaviors (amount patterns) indicate fraud more than the amount of a single transaction. The implication of these studies is that relative amounts (e.g., compared to a customer's average spending) would be more meaningful.

From a statistical standpoint, hypothesis tests have been introduced to compare frauds and non-frauds. For example, West and Bhattacharya (2016) highlighted the dearth of focused studies on the effect of transaction amount and suggested deep statistical exploration. Traditional tests such as the t-test or the non-parametric Mann-Whitney U test can fortify the argument that fraud transactions tend to weigh with amounts that average higher. These tests must consider the underlying population distribution, indicating that the transaction amounts are non-normally distributed and highly skewed (this distribution will often exhibit a long tail of large values!). To this end, this project will check the significance of the difference in amount distributions using the Mann-Whitney U test (which does not assume normal distribution), hence adding statistical rigor to the ML findings along with a non-arcane interpretative conclusion (e.g., "fraud amounts are significantly greater than non-fraud amounts at p<0.001").

Hybridization of statistical formulation and ML has also been extensively employed. A meaty practical system is Bank Sealer (Carminati et al., 2015), which combined rule-based analysis, anomaly detection, followed by disclosing results via a visual interface to investigators. It brings to bear yet again that automated algorithms will often be just one part of a larger fraud detection ecosystem, which will include business rules (for known fraud patterns), anomaly flags (for novel patterns), and human expert review. The ceramic spirit of this approach has inspired subsequent research: Lee et al. (2018) developed EVA (Expert Visual Analytics), which allows human analysts to explore features such as transaction amount distributions in fraud identification. These interactive systems leverage statistical charts such as histograms and box plots and simple machine learning, allowing experts to adjust detection rules and effectively close the feedback loop between algorithms and human insight.

Another evidence of hybridism is presenting topics in unsupervised-supervised combination in literature. For example, there are clustering algorithms such as k-means and DBSCAN that collect transactions into behavior clusters and that get then labeled or fed into the classifier. Multi-stage fraud detection pipelines might be the topic of discussion by Mahdavifar and Ghorbani (2019) (as reviewed by Al-Hashedi & Magalingam). Stage 1 might rule out obvious non-fraud; stage 2 might apply an ML model to the now remaining ambiguous cases, while stage 3 might flag anomalies on an ongoing basis. Al-Hashedi and Magalingam reviewed how hybrid approaches: statistical + ML became more popular over 2009-2019 as it may remove the limitations inherent in any one of the approaches. For example, an ML model may be exactly accurate but could miss types of fraud it was not trained on - an anomaly detection method running in parallel might catch those novel cases.

Imbalanced learning techniques themselves can be seen as a hybrid between data-level statistical adjustment and algorithm-level changes. SMOTE, mentioned earlier, is a statistical oversampling technique that frequently combines with ML models in fraud research to counteract skewed class distribution. Variants like ADASYN (Adaptive Synthetic Sampling) further automate the oversampling of harder minority examples. These methods are not specific to fraud, but they have been shown to improve fraud classifiers when used with caution (too much oversampling can lead to overfitting the synthetic data). Studies such as Misko et al. (2020) (in some financial fraud contexts) compare oversampling with alternative approaches like one-class SVM or isolation forests (which model the majority class normal behavior only). The consensus is that one size does not fit all; thus, ensemble and hybrid strategies have been observed to be the best solution in most cases. For instance, a fraud detection solution can simply ensemble a supervised booster (known patterns) with an unsupervised outlier detection (anomalies) to achieve better coverage.

In effect, the literature shows that both machine learning and statistical domain insights are robust in practical fraud detection, showing that statistical insights into the domain help facilitate the application. Predictive modelling is provided by machine learning to manage the complexity of interaction with hundreds of features while ensuring that statistical suggestions and expert insight breathe life into the findings (like ensuring that frauds show different spending patterns) and assist with system tuning (like review policies based on thresholds). This project leans on these insights: we perform statistical analysis to tackle a domain question (the role of transaction amount) and predict using the latest techniques in ML, with a view to a hybridist understanding of the problem.

## 2.4 Gaps and Current Research Trends

Despite considerable efforts, several holes persist, a few of which our work fills. One of them is that, despite the many studies that include the amount as a feature into the study framework, they do not elaborate much on its specific standalone effect. Our statistical analysis in this project directly measures the relationship by bringing much-needed empirical evidence on this issue. Second is how interpretable and actionable fraud models typically are. Most recent works have put forward models calling for deeper or ensemble consideration, while most common would-be models require an apparent reason for that credit or debit not being allowed (for purposes of customer communication or regulatory compliance). This is achieved by SHAP, using this to interpret model output and introduction of terms along which features like amount, device, or address match contribute to prediction of fraud.

Another gap is how to manage the dynamic nature of fraud (patterns can shift quickly). Though this project adopts a historical static data set, in discussion we consider model updating and the concept drift issue. Literature suggests solutions like online learning algorithms or periodic retraining with recent data to maintain performance. We also note the emerging trend of federated learning for fraud detection: banks are exploring how to make their models learn together on one-wide fraudulent pattern across banks without exposing customer data. This is beyond the scope of our project, but our results could be integrated into a federated scenario in the future (e.g., sharing only model gradients, not raw data).

Finally, there is the deployment challenge, which is however less studied academically: how to plug ML models into online screening systems in which the constraints of time latency are very tight and involve retrofitting with legacy rules. To understand how fast a fraud detection model needs to decide, we can imagine it has milliseconds to do so during a credit card authorization. Most of the papers do not explicitly address these practical issues, which are vital, however. The issues addressed in the discussion of our project touch on the runtime performance in terms of the above points on how to balance complexity with speed.

In conclusion, this literature from 2018 to 2024 makes a vibrant field combining advanced machine learning to classical techniques for enhanced detection of fraud. Our work takes up learning from these works – using tried and tested algorithms (XGBoost/LightGBM), even conservatively oversampling, attempting to include interpretability – but also fills specifically the niche void of understanding transaction amount's role. That will be followed in detail by the methodology that operates these ideas from the IEEE-CIS dataset.

# Methodology

This section elaborates on the data and tools employed, their data cleaning and exploration, machine-learning modeling approaches (with validation strategy and evaluation measures), and finally, statistical testing of transaction amounts. We aim to set our methodology in a way that mirrors the developing cycle of a realistic fraud detection system: raw data arrives, data cleaning and transformation occur, insights are obtained through EDA, prediction modeling is built and tuned, and finally, the results are interpreted and verified through statistical tests and some explanations.

## 3.1 Data Collection and Integration

The main data source for this project is the IEEE-CIS Fraud Detection dataset: this dataset was first made available for use in an IEEE Computational Intelligence Society Kaggle competition in 2019. The data consists of two sets of files, one containing transaction-level data (financial transaction records) and the other containing identity data (digital authentication data). Here, we focus on the training part of the dataset, which has transactions amounting to 590,540 having a unique identifier, TransactionID. Of these, 20,663 transactions are labeled as fraud (isFraud=1), which is about 3.5%. Each transaction record has features like transaction timestamp (TransactionDT), transaction amount (TransactionAmt), product code (ProductCD), payment card data (card1 through card6), address information (addr1, addr2), email domains (P\_emaildomain, R\_emaildomain), and others up to TransactionID 590,540. For identity data of selected transactions, this contains attributes like a device type, operating system, browser, and the various digital identities fingerprints (id\_12-id-38). For the sake of privacy, many of our feature names are anonymized (example, V1-V339 for engineered features and masked device information). A significant first step was joining the transaction and identity datasets on TransactionID to yield a single-integrated table for analysis, as advised in the instructions provided with the dataset documentation.

We left outer joined identity data to transaction data using TransactionID as the key, resulting in a merged dataset composed of 590,540 rose by 434 columns (including the fraud label). Not all transactions have identity records, approximately 144k transactions have identity information, meaning identity-related features will have missing values for many entries. We keep all transactions in the analysis to avoid bias towards those with identity data; missing identity features will be dealt with in preprocessing. The separate test provided on Kaggle was not used for model evaluation because it has no labels; thus, our training data are split for validation as described later.

## 3.2 Data Preprocessing and Cleaning

Preprocessing the dataset is critical for the present study because it considers the substantial number of records it contains (hundreds of thousands) and its high dimensionality (more than four hundred features, many of which are sparse or high cardinality categorical). The memory optimization preprocessing pipeline included the handling of missing values, feature encoding, and then scaling or transformation.

### 3.2.1 Memory optimization:

Loading the raw CSVs into pandas, the dataset uses over a few gigabytes of memory. We downcast numeric columns to more efficient dtypes (e.g., from 64bits floats to 32bits) and converted some categorical text fields to categorical dtype. It decreased memory consumption significantly (> 50%), which is important for enabling faster processing and model training with the usual hardware.

### 3.2.2 Missing Values:

A few features help identify individual transactions as not captured by the identity database or are applicable to a type of transaction (like some id\_ features). In this scenario, we calculated the percentage of missing values for each column. Under the principle that extremely sparse features will not be reliably informative, we apply the rule of dropping any column that is greater than 50 percent missing. This resulted in dropping some high id\_ features and some V features that were mostly empty. The rest were treated as such to impute missing values column-wise. Column-wise imputation was done for all entries where the features were numeric (e.g., transaction amount, or numerical device info), replacing the missing entries with the median of that feature (to avoid skewing influence that could arise from the mean). Categorical features (e.g., card type, product code) were replaced by the mode (most frequent value) of that feature. We made sure there was no missing data after that simple imputation. For example, if card4 (that would be the card network) was missing against some records, we would fill that with "visa" if that was the most common category. Similarly, missing TransactionAmt were rare (if any), but would be filled with median transaction amount. In addition to this, we do note that imputation could distort distributions (especially in cases of very indicative features) but considering that it had low missing ratios for key features and that it uses robust tree-based models, such compromise lies within what could be called a reasonable compromise between simplicity and performance. In addition to that, we also thought of bringing up an explicit "missing" category flag for those features for which absence could be informative into itself (like with id\_ features which mean "no identity info available") but rather decided to let the imputed placeholder be treated appropriately by tree models.

### 3.2.3 Feature encoding:

The dataset has categorical variables of various kinds: real categories like ProductCD (a product code such as W, C, etc.), card4 (card network: Visa, MasterCard, etc.), and card6 (debit/credit indicator). Also, nominal IDs like card1 (which appears like an anonymized customer identifier or card identifier with thousands of unique values) and P\_emaildomain (purchaser email domain). For low-cardinality categoricals (such as product code with five levels, or card4 with four levels), one could use one-hot encoding. However, high-cardinality features such as card1 that have >10k unique values and email domains > 50 unique would cause one-hot encoding to explode dimensions of features. Therefore, we preferred to use label encoding/frequency encoding for categorical features. In real practice, tree-based models like XGBoost and LightGBM are capable of handling categorical features in one-hot formats but since we won't use inherent orders for these, we often employ frequency encoding (replace each category with the frequency of that category in the dataset) to give a sense of prevalence. For instance, card1 converted to how many times appeared in the data each card ID (this essentially provides a feature of "card usage count," which may be informative - few cards (IDs) appear very frequently, indicating either a popular card or a bot using that card repeatedly for fraud). Similarly, email domains were encoded by frequency; rare domains might be more suspicious. For card4 (networks) and card6 (debit/credit), we label-encoded them to integers since they have few levels. These gradient boosting algorithms can easily handle categorical splits if they are passed as categorical dtype (CatBoost) or can find splits based on the encoded numeric values (XGBoost/LightGBM). So, all features are converted in numeric form with no need for explicit one-hot expansion, except for internal restriction cases.

### 3.2.4 Feature scaling and transformation:

Tree models are not sensitive to monotonic transformations or scaling of features. Neural network models, on the other hand, do benefit from scaled input ante. The log transformation was also applied to the TransactionAmt feature because of much skewness in transaction amounts (ranging from exceedingly small to large purchases with a long tail). Log1p (natural log of (amount+1)) transformation was applied to TransactionAmt. It improves Gaussian distribution while reducing outlier impacts. Thus, this behavior would stabilize the training for the neural network and can marginally help tree models consider proportional differences. Other numeric features such as counts or intervals were also considered for transformation; however, TransactionAmt was the primary prétendant in need of scaling. It would not be possible to standardize (z-score) the features on a global scale since three models do not require it and for neural network which we manage separately if the need arises.

After these steps, we dropped identifier columns that are not used for modeling (TransactionID was dropped after merging, and TransactionDT – a relative time stamp – was transformed into useful features like day of week or hour, described in EDA below). The output of preprocessing was a clean dataframe ready for analysis and modeling.

## 3.3 Exploratory Data Analysis (EDA)

This was here, prior to any model building, that Exploratory Data Analysis was performed to understand the data characteristics and what differentiates patterns of fraudulent and non-fraudulent transactions. EDA is utilized to satisfy curiosity over questions like the amount versus fraud relationship, and it further provides insights into feature engineering for the modeling stage. Some important aspects of our EDA included:

**Univariate distributions**

It involved an analysis where we looked at the distribution of transaction amounts for fraudulent versus non-fraudulent transactions. This involved plotting histograms and density curves for TransactionAmt for two classes of input variables. Given the skewed nature of amounts, we plotted them on a log scale or used density estimation to view class-wise differences in shapes. We also assessed basic statistics such as: mean, media, and quartiles for each. In looking at categorical features, such as ProductCD or card4, we also checked for higher fraud rates across certain reference categories (e.g., does product code "W" have disproportionate fraud?). Bar plots of fraud rates by category were employed.

**Bivariate analysis and correlations**

In scenarios A to B, we examined things like TransactionAmt vs. TransactionDt (time) to check whether fraud would cluster in certain time periods, and TransactionAmt vs. card features (like do excessive amounts get associated with a particular card type or bank). Additionally, we prepared correlation matrices for the numeric features to detect if some engineered V features are highly correlated to amount or to each other (that would mean redundancy). It was difficult to use many V features since they were anonymous, hence we ended up focusing the EDA on interpretable features.

**Time-of-day and Temporal Patterns**

We took some extractions such as hour of day or day-of-week feature from the transaction time, (TransactionDT being a numeric timestamp offset). Herein, a hypothesis existed that fraud might be high during some times (e.g., midnight hours) when criminals do transactions in the hopes of the user being asleep, etc. For this, the density function of transaction hours plotted nationwide for fraud versus non-fraud was used. What was indeed interesting: Fraud transactions occurred more during early morning hours (around 2-3 AM in the dataset's time units), while genuine transactions had their peaks during business hours, as already shown in Figure 1. This fits domain-wise and was our first hint that time could emerge as a useful feature, if not for rule-based detection.

**Fraud rates by category**

Using categorical features, we calculated fraud rates per category. So, for each card network, i.e., card4 being Visa, MasterCard, etc., what percentage of total transactions were fraud cases? This distinguishes the riskier category. One such indicator of concern might be if there were significant deviations between networks concerning fraud rates (sometimes, user demographics in that network could be a confounder). From what we observed though, fraud rates are equal among the major credit card networks, with an uptick for Discover and American Express compared to Visa/Mastercard, this might come down again to their sample sizes or an established pattern against them. Product codes also showed differences in fraud incidence; code "C" presented with a higher fraud rate than "W", an indication that whatever is in "C" might be either purchase of digital goods or high-risk goods.

**Address and email match.**

The dataset features like P\_emaildomain (purchaser's email) and R\_emaildomain (recipient's email, for shipping) were used to check whether mismatches (purchaser vs. receiver domain differently) correlate with fraud. Also, the address fields addr1 (billing region) and addr2 (most probably country) were evaluated: Foreign or extremely far addresses may signify fraud (mismatch between IP location and billing address). Some free domains have a slightly larger fraud rate (e.g., emails from less-known providers may be disposable addresses for the fraudster). Fraud incidence was lower if the purchaser email and recipient email belonged to the same domain (which may mean item is delivered to the buyer).

Some ideas were confirmed by EDA, which provided quantitative evidence for some correlations between transaction amount and fraud. We found fraudulent transactions to have, in general, a higher median amount than non-fraudulent ones. For our analysis subset, the median Transactions for fraud were about $120, whereas for legitimate transactions it was $70 (note: we use "$" loosely since units of actual currency are anonymized). The distributions (Fig. 2) tell us fraud transactions and non-fraud transactions are alike in that mostly the transactions worth less than $200 account for the bulk of both types; however, the fraud distribution has a heavily skewed tail-in other words-more high-value frauds. Less than $500 transactions are mostly frauds, even though statistically, they are rare. This EDA finding provides a preliminary answer to our research question: indicating larger amounts are associated with a greater chance of fraud, but there is considerable overlap (there are lots of small frauds and many large legitimate purchases). So, the amount will not be a great predictor by itself, but it will provide a signal.

A graph of a distribution

AI-generated content may be incorrect.

Figure 1

*Figure 1: Fraud vs. Non-Fraud Transactions by Time of Day.* *Density plots of transaction volume over the hours of the day for fraudulent (orange) and non-fraudulent (blue) transactions. Fraud tends to be more common during late night to early morning hours (peaking around hour 2-3), whereas legitimate transactions have higher density during daytime. This suggests time-of-day can be a discriminative feature for fraud detection.*

A graph with blue and orange lines

AI-generated content may be incorrect.

Figure 2

*Figure 2: Transaction Amount Distribution by Class (Log Scale).* *Overlaid density histograms of transaction amounts for fraud (orange) and non-fraud (blue). Both classes are highly skewed to the right (plotted on log scale for clarity). Fraudulent transactions show a slightly higher median, and a heavier upper tail compared to non-fraudulent ones. While most transactions in both groups are low-value, fraud is overrepresented among the largest transactions.*

The data distribution differences were evaluated quantitatively using statistics (§3.5), in addition to visual examination. In EDA, we also noted issues with the quality of the data: for example, we observed a few negative values or zeros in TransactionAmt (these could either indicate refunds or zero-dollar auth holds) and a few extremely high values (outliers). We felt it was essential not to remove any outliers since it would also remove fraud cases (some frauds, after all, could be extreme amounts). But even though we opted not to remove these values, the log transformation would reduce their influence. We also learned that TransactionDT is a relative timestamp (in seconds relative to an unspecified start), which we computed to generate human-readable features (hour, day) rather than inputting into our model in raw form, as doing so might have resulted in data leakage (since it gains a second with each transaction, data cuts by time are done cautiously, as we will see in our validation split).

Insights from the EDA permitted some features to be engineered: these included hour of day, weekend vs during the week in a boolean matrix, and whether billing and shipping addresses match (via addr and P\_emaildomain and R\_emaildomain). These features were added to the modeling dataset. Also noted were the most promising variables (TransactionAmt, frequency of card1, email domain, device information like DeviceType and DeviceInfo). With that understanding, we moved to model training.

## 3.4 Machine Learning Modeling Approach

We framed fraud detection as a binary classification problem, predicting the isFraud label (1 or 0) for each transaction. Our modeling strategy consisted of training and evaluating several algorithms, comparing their performance, and choosing one for the final analysis. The main algorithms we explored were: Gradient Boosted Decision Trees: The three boosting frameworks that were utilized were XGBoost, LightGBM, and CatBoost. These algorithms have been shown to perform well in the structured data tasks, thereby managing a mixture of numeric and categorical features with excellent balance. Also, they provide effective ways to manage the imbalance in the dataset (scale\_pos\_weight or tuning parameters), and high accuracy is always obtained from those algorithms in real-life fraud data sets. Hence, we expected them to perform well with extraordinarily little scaling needed with these features.

**Neural Network**

As a baseline deep learning technique, we considered a simple feed-forward neural network (multi-layer perception). In the case of our network, it took the pre-processed features with appropriate normalization and had a couple of hidden layers with ReLU activations. Owing to the imbalanced nature of our problem, we considered using either a weighted loss (larger weight for fraud class) or to perform resampling during training. There was some expectation that the neural net could require significantly more data and tuning to achieve the same level of performance as the boosters but could potentially model interactions in a manner different from them.

**Logistic Regression**

We performed simple tests with logistic regression on a reduced set of features (after one-hot encoding some of the major categories and scaling), for the purpose of constructing an interpretable benchmark. Its performance, however, was expected to be exceptionally low, so it served as a metric against which to measure how much more the complex models gained.

**Our training process for the ML models was as follows.**

In this work, we focus on the important functioning of imbalanced classification, where frauds are a tiny minority. Out of 780,000 labelled transactions, only 3.5 percent are detected as fraud. After skipping feature engineering for a while, we looked at the feature distributions and considered if there were any immediate data issues considering fraud seems to be so rare. We could describe five hundred features divided between transaction attributes (60 features) and identity attributes (many of them categorical). Afterward, a classification is made between hinge, lin, and relu; and some results are obtained.

**Train-Validation Split**

For every model we built, we used a different apparent method of forecasting fraud actions, using training datasets from one set and testing on a new one. For example, if transaction DTs are not entirely different, the model may be infusing itself with information from the validation set too. Time-based split testing gives more preference to heavyweights, analyzed and well-situated prediction. On this note, we acknowledge that the entire scenario of production testing would aim at assessing the model with real-life future data; hence we could exploit random stratified split for training. We create an unequal split, adding even more confession to its bias.

**Cross-Validation**

Here, we apply hyperparameter tuning and try to gain a robust estimation of model performance by performing stratified k-fold cross-validation with k=5 folds in the training set: that is, with CV, we divide training data into 5 folds, each serving, in turn, as a mini-validation fold while training on the other 4, Stratification ensures that fraud ratio is preserved in each fold. The main use of CV was hyperparameter search (especially for NN and then to verify consistency of results across boosters). Performance on these folds, as measured by ROC AUC and recall/precision on the fraud class, was used to adjust parameters.

**Hyperparameter tuning.**

We conducted a grid search (some manual tuning also) for each of the models:

* XGBoost/LightGBM-Tuning of critical hyperparameters includes learning rate (tested values around 0.01 to 0.1), maximum tree depth (6, 8, 10), number of estimators (maximum of 500 trees trained with early stopping), subsample ratio (0.6-0.8), and colsample\_bytree (between 0.3-06 to prevent overfitting taking into consideration the large number of features). We tuned the scale\_pos\_weight to tackle the problem of class imbalance (which was set at 1:29 to account for 3.5% fraud; this gives the booster extra focus on fraud instances). Given its superiority in handling categorical features, we also specify which features are categorical for CatBoost and tune it similarly for depth (6-10) and learning rate.
* Neural network-Furthermore, we changed architecture (varying 1 vs. 2 hidden layers with 64 to 128 neurons), tuning learning rate for the adaptive moment estimation optimizer (1e-3 to 1e-4), tuning batch size since we have class imbalance problems, we preferred smaller size ~256 to ensure that in each batch there are some fraud examples), and tuning epoch with early stopping based on validation loss/AUC. Also, for every epoch, we tried oversampling the fraud class such that more fraud examples were introduced to the network and using class-weighted loss putting a higher weight to the fraud.
* MaxDepth=6, eta=0.05, nEstimators=500 (with early stop), subsample=0.8, colsample\_bytree=0.4, and scale\_pos\_weight≈29.

**Training**

The best hyperparams were applied to training the models based on the training set. For boosters, we used early stopping with a patience of fifty rounds in the 20% validation split to guard against overfitting (monitoring AUC on val). For neural networks, the training continued for up to 50 epochs unless a much earlier shutoff point was reached earlier when validation loss plateaued.

**Evaluation Metrics**

Given the imbalance of our data, we opted for ROC AUC (Area Under the Receiver Operating Characteristic) as the major metric to guide model selection, with AUC being unaffected by class ratio and the ranking performance (that is, how well the model discriminates between fraud and non-fraud). This is sensible if we consider whereby in fraud detection, we are interested in catching fraud and so do not overly care about "how much" a non-fraud is few as it were-another (false positives). Besides AUC, precision, recall, and F1-score are examined for the fraud class and relate to the selected probability threshold (default 0.5 or adjusted). High precision rates for low false alarms should reduce operational costs, and high recall rates for false positives will increase loss savings. The Precision-Recall curve is also looked at. In our case, AP for approximately 5% is similar across all models. This PR curve is useful when concentrating on performance for the minority positive class. Finally, even though we report overall accuracy in the 96.5% post baseline return (if predicting all non-fraud), given the metric above, it is not highly informative regarding the model's quality. For clarity of access, the confusion matrix for the threshold assumed to balance precision/recall gives a sense of how false positives and false negatives count results.

**Imbalanced data handling**

We also introduced some SMOTE oversampling as an alternative to using class balancing techniques such as scale\_pos\_weight in XGBoost/LGBM or class weighting in NN. More specifically, SMOTE was executed to create more minority class examples at around 10% of the training data (from 3.5% originally). During our experiments, SMOTE has given marginal enhancement to recall for simple models; although, once hypertuned, not much benefit was seen in the instance of XGBoost (most probably because the booster had already made provision for managing imbalance and then on top of this, we could have introduced the threat of overfitting owing to the inclusion of synthetics). It was decided that SMOTE would not be adopted in the final model pipeline. Instead, it was determined that the internal methods of algorithm handling and threshold tuning should be relied upon. Notwithstanding these reasons against SMOTE, we have reported in the interest of discussion, as it is a commonly referred accentuating method in literature, and, indeed, based on one's strategy one does use SMOTE or the newer ADASYN in combination with certain classifiers.

**Model selection**

We were able to start with the AUC and recall indices of the models compared on the validation set after training. It was found that tree-based models were better than neural networks. Specifically, it looked like LightGBM and XGBoost had almost equal performance levels (AUC ~ 0.95)—just slightly higher than CatBoost (AUC ~ 0.94), due to the fact we tuned the other models more carefully. Neural networks did achieve an AUC of around 0.88 after turning; however, there was greater instability with them, leading to decreased precision (therefore more false positives) except when we set a high threshold. With the results mentioned earlier, the XGBoost model was selected for further evaluation and interpretation due to having just that edge over quality coupled with extremely high precision. LightGBM results are also given for comparison. Techniques to ensemble various models (say, through averaging XGBoost and LightGBM predictions) were explored and they did improve the AUC in a trivial manner (~0.956 versus 0.953), hardly enough to justify the additional complexity though during a competition it might be worth the effort to ensemble for every bit of AUC. Therefore, for the sake of simple explanation, only one model is addressed.

## 3.5 Model Interpretation (SHAP) and Feature Importance

Understands quite how much our model can tell us about why it raises an alarm on certain transactions, or in other words, can be seen as holistic consideration, which dovetails with the raw performance measurement. To inspect the SHAP (SHapley Additive Explanations) and decipher feature contributions to our best model, we invoke this method. Introduced by Lundberg & Lee (2017), SHAP assigns each feature a Shapley value which represents its contribution to pushing the model output away from the model output mean for one given instance. This gives global importance to features (by aggregating absolute SHAP values throughout the dataset) as well as local explainability (which features pushed some transaction to be fraud or not).

For SHAP value calculations, a small portion of the validation set was used because of computation time on 400+ features. The output then gives us a ranking of the importance of features and a visualization of how feature values relate to the predicted outcome:

**Global importance**

As per SHAP analysis, high-impact features affecting fraud prediction involve TransactionAmt (log-transformed) and frequency of card1 (frequency of that card ID's appearance in the dataset, proxies whether the transaction comes from a card that has been highly used in the dataset), DeviceType/DeviceInfo (whether it was a mobile device or PC, and if so what specific browser user agent), an address match flag (derived addr\_match), and some of the anonymous features which likely capture transactional patterns. Notably among the incredibly notable features, an identity indicator (for example, id\_20) and the email domain were indicated. This result tells that the model is relying on a mixture of transaction characteristics and digital fingerprinting features.

**Card usage (card1 frequency)**

The number of transactions related to the same card1 (an encoded feature of our own) has been flagged by the model as an important parameter. The SHAP observations would suggest that the transactions from cards that appear extremely often on the data used were more prone to fraud - perhaps because fraudsters using a stolen card might make many attempts in a short span of time, or some of the card ID corresponds to test accounts which generate lots of transactions. Just as well, there existed some fraud risks as well for cards with very few transactions (one-off high fraud attempts). In terms of the model, medium frequency cards were thus the safest. SHAP could allude to such U-shaped behaviors (both exceptionally low and extremely high frequencies being risky) by illustrating the ends of the scale for the feature in question and its SHAP effects.

**Device Type & Browser**

DeviceType included (desktop or mobile), found that slightly leaning on desktop transactions would be classified as fraud: most transactions were mobile so fraud would be through PC scripts. Red (e.g., positive toward fraud) displaying the average with small magnitude then blue for mobile as well showed for DeviceType SHAP values. The browser possessed interesting patterns: very outdated browser versions or rare browsers had positive SHAP (fraud indicator), since legitimate users tend to use common browsers, whereas fraud automation might use headless or old browsers. This insight might help with rulemaking (for instance, flagging transactions with uncommon user agent strings).

**Address match (billing/shipping/email)**

The derived flag whether it matches between billing and shipping had an impact quite vast. When the addresses or emails are mismatched, SHAP is positive (higher chance for fraud) and supports the statement that most often, fraudsters will ship to a different address or will use another email than that one to which the card is attached. The opposite would then happen when everything has matched up; then, the SHAP would be negative (counting down from the chance for fraud). It was a normal feature of importance about what is used as anti-fraud checks in the industry in general.

**Other top contributors**

Incredibly important raw anonymized features in analysis Vxxx, raw feature importance by gain in XGBoost for instance in V258 and V201 (just to mention examples). We do not know what this means (because a lot aggregated some transaction info), and we rely on SHAP to see at least the effect: by pattern it would seem some kind of derived relationship with the card or time. We will not speculate very much, but even not knowing what they are, it is worth mentioning that including them helped the model, which thus achieved high AUC. In a production scenario, one might exclude such opaque features for interpretability or retain them if they consistently help and use SHAP to partially interpret them (by correlation with known factors).

A graph of different types of data

AI-generated content may be incorrect.

Figure 3

*Figure 3: SHAP Summary Plot of XgBoost Model Important Features. Each dot is a SHAP value for a specific transaction for a specific variable (y-axis). Blue dot indicates a low level for that feature, red dot indicates an elevated level. Hence the value holds the effect these changes would have on the output of the model (SHAP value, where positive means pushing toward fraud). High TransactionAmt (red on right for that variable) points toward a prediction for fraud, whereas a minimal amount (blue on left) pushes in the direction of non-fraud. With respect to the card usage count, extremely high counts (red) increase the risk of fraud but also incredibly low counts to a lesser extent (blue on right), suggesting that both unfamiliar and overly prolific cards give rise to suspicion. Distance (proxy for geo-distance between billing and IP) if large (red) raises the chances of fraud. An "Address Match" being false (encoded as one, red) also pushes towards fraud, while it goes to non-frauds when it becomes true (blue). This helps the model's decisions be better understood with stakeholders.*

In addition to the overall explanations, SHAP was also utilized for the interpretation of example transactions for local explanation, as in a case study (for detailed cases, refer to Appendix C). For instance, in one such transaction, in which the model flagged the suggested observation as suspicious with high confidence, we were able to demonstrate that the combination of a high amount, mismatched shipping address, and an unusual email domain accounted for the highest contributions toward such prediction (each contributing, say, +0.3 to the log-odds of being fraud), although the usage of a mobile device reduced that slightly. These detailed disclosures for individual cases, however, are very vital to an investigator who is looking into giving the model's output credence: it said "fraud," but better than that, it will say, "amount is $900 (high), shipping to different state, and using an odd email provider."

## 3.6 Statistical Hypothesis Testing

To thoroughly address the research questions regarding transaction amounts and the likelihood of fraud, we conducted several statistical hypothesis tests. We compared the amount distributions of fraudulent transactions with those of legit transactions. The null hypothesis was that both groups - fraudulent and non-fraudulent transactions - have the same distribution and that any difference in sample means or medians is due to random effects. The alternative is that such distributions differ (specifically, we expected higher central tendency for fraud).

Because the amounts which are non-negative, right-skewed, and heteroscedastic lead to skewed distributions, Mann-whitney U test (or Wilcoxon rank-sum test) became our primary test. This non-parametric test does not assume normality and assesses whether one distribution tends to have larger values than another instead. It ranks all data, comparing rank sums. We have Welch's t-test also at the disposal for comparison where unequal variances (the division suits this case owing to a high variance of fraud amounts) exist on transposed logs since it assumes no equal variances.

The sample size of this test was very large (n\_nonfraud ≈ 569k, n\_fraud ≈ 21k after preprocessing), ensuring that even truly minor difference can be significantly detected. Indeed, it showed a U statistic that was quite far from the null expectation and. p < 10^-10 (0) - which implies the existence of a statistically significant difference. More informative is the effect size: we evaluated the median of each group - median(non-fraud) ≈ $50 and median(fraud) ≈ $120 in the raw data (these are rough numbers due to anonymization and scaling). The mean was also higher for fraud amount (fraud mean $160 vs non-fraud mean $110, again approximate). Although statistically significant, these differences would not be phenomenal from the perspective of practice and yet believe confirmed that frauds tend to involve higher amounts. Also, using the Mann-Whitney test estimates the probability that a random fraud transaction has a higher amount than a random non-fraud transaction; that probability was ~0.58, which means that there is a 58% chance a fraud pick will have a higher amount than a non-fraud pick - above the 50% null baseline, but not close to 100%. In short, this quantifies that amount is a contributory and not a dominant factor.

There also were no significant differences found through Welch's t-test on log(amount), p <0.001, which showed higher log-mean with respect to fraud transactions. In addition, these transaction amounts were binned into categories and fraud rates calculated for each: transaction value of < $1; $1-$10; $10-$100; $100-$1000; >$1000. Incrementally, these fraud rates rose with higher bins- from <1% in the smallest bin to >10% in the highest bin- thus reinforcing this association.

This can be summarized simply that the statistical tests hold true for this dataset in a sense. Higher transaction amounts are associated with the higher incidence of fraud. This agrees perfectly with our earlier exploration data analysis and with the model's use of that feature. However, because of overlapping distributions, amount works along much tighter lines. It is not predictive enough to be a rule. Most frauds are small; quite a few large transactions are genuine. It comments on the wealth of reasons why there is a need for a multivariate ML model. Our findings here align with some threshold-based rules: for instance, banks might apply more stringent verification for transactions above a threshold amount because they tend to have much higher rates of fraud risk (not just block them). Having presented a methodology dealing with data, modeling, and statistical testing, we now go on to the results section, presenting model performance, examples of detection, and their interpretation in the problem context.

# Results

In this section, we present the outcomes of our experiments: insights from exploratory analysis, performance metrics of machine learning models, evaluation of the best models, and feature importance analysis. We integrate visual results to illustrate key points and organize results into subsections for clarity.

## 4.1 Exploratory Analysis Results

The Exploratory Data Analysis revealed several patterns and insights:

* **Class Imbalance**: Only 3.5% of transactions in the training dataset were fraudulent, with 96.5% being legitimate transactions. This significant imbalance presented a challenge for model training and evaluation.
* **Missing Values**: The dataset contained a high percentage of missing values, especially in the V columns. Around 414 features contained missing values, requiring strategic handling during preprocessing.
* **Device and Browser Patterns**: Fraud transactions showed higher prevalence in certain device types. For example, fraud was more prevalent in mobile devices and in transactions using 'IP\_PROXY' based on 'id\_31'. This suggests fraudsters might prefer specific device configurations or attempt to mask their identity.
* **Temporal Patterns**: Fraudulent transactions were not uniformly distributed over time. The data showed that 'TransactionDT' represented a timedelta gap rather than an actual timestamp, allowing us to extract temporal patterns by converting it today/week/month units.
* **Data Distribution**: Many anonymized columns had non-normal distributions, requiring careful feature engineering and transformation approaches.

## 4.2 Model Performance and Benchmarking

We evaluated multiple gradient boosting models, with the following results on the validation set:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Public Score | Private Score |  |
| LightGBM | 0.961445 | 0.938790 |  |
| LightGBM v.2 | 0.952711 | 0.928091 |  |
| XGBoost | 0.959648 | 0.935475 |  |
| CatBoost | 0.958168 | 0.932944 |  |
| Ensemble (0.8 LGBM + 0.2 CatBoost) | 0.963440 | 0.941706 |  |

The gradient boosting models were shown to outperform the simpler ones by large margins; in our cross-validation studies, XGBoost achieved an AUC of approximately 0.957, whereas LightGBM, also scoring exceedingly high with an AUC of approximately 0.942. Both models utilized early stopping to avoid overfitting and ensure practical training.

The winning method was an ensemble of LightGBM (80%) and CatBoost (20%), whose public leaderboard score attained was 0.963440, while that of the private leaderboard was 0.941706, ranking it a gold medal.

Each of the individual models was put through 5-fold cross-validation to ensure robust evaluation of their performances. LightGBM was trained with 0.7 subsample, 0.7 colsample\_bytree, and a learning rate of 0.005. XGBoost was trained with 0.01 learning rate, max depth of eleven, and 0.8 subsample. Both performed very well in prediction with AUC values above 0.94.

Some discrepancy was observed in our public leaderboard (0.961) and private leaderboard (0.938) scores when considered. This suggests that there was some overfitting in the public leaderboard. Thus, extra tuning and validation approaches must be adopted to enhance generalizability for the model.

## 4.3 Feature Importance and Model Insights

Statistical feature importance by means of LightGBM revealed key patterns:

**Device and Identity Features:**

The device and identity-based features were highly significant, with great strength in their contributions. The id\_split function, which collected DeviceInfo, id\_thirty, and id\_thirty-one columns and thus extracted information such as device name, device version, operating system, and browser ID, added to the prediction power of the model.

**Temporal Features:**

Temporal-based features extracted from TransactionDT—like DT\_D, DT\_W, and DT\_M, which were days, weeks, and months respectively, from a start date—were important in capturing the seasonality of fraud patterns.

**Store and Product Features:**

For every ProductCD category, transaction counts calculated in mean fraud rates across windows of time were useful. These features (ProductCD\_W\_Day, ProductCD\_C\_Day, etc.) captured useful patterns related to product-specific fraud and frauds over time.

**Card and Identity Hashing:**

uid1 (card and address features combination) and device\_hash (hashing device features) engineering features were vitally important to tracking unique users and devices. It was useful in discovering repeated fraud schemes across devices and users.

**Categorical and Numerical Transformations:**

The titular treatment of categorical columns through label encoding and elsewise has empowered the model through the addition of various transformations, including counting occurrences across categories, amending of D series features, and interaction features.

Good agreement in model performance indicates that these features and engineering techniques have collectively solved the challenges of targeting fraud patterns peculiar to this dataset.

## 4.4 Challenges and Limitations

In undertaking this project, we experienced several challenges:

1. **Dataset Sparsity**: This problem was so severe that most feature engineering and selection processes could be perceived as noise obscuring the desired signal.
2. **Missing Values**: There are 414 features with missing values, and we opted to assign NaN values by substituting them with ridiculously small values (like -999) instead of deleting the information.
3. **Class Imbalance**: The model evaluation and threshold selection processes required extra consideration when the fraud-to-non-fraud transaction ratio of 3.5% versus 96.5% was considered as a significant class imbalance.
4. **Outliers**: The outliers require extra measures by replacing the values that rarely occur with some predetermined value (like -9999).

## 4.5 Implications for Fraud Detection

The implications of the discussed results for real-life applications of fraud detection are:

1. **Feature Engineering Impact**: The detailed analysis of the feature engineering processes on time-series data, device hashing, and categorical interaction features served in enhancing the model performance.
2. **Model Selection**: The second most important finding is that both LightGBM and XGBoost performed equally well, but XGBoost outperformed LightGBM by a slight margin in AUC. The best generalization was achieved by the ensemble approach of LightGBM and CatBoost, proving model diversity.
3. **Cross-Validation Consistency**: The AUC scores obtained in the cross-validation folds indicate that both primary models had a similar performance, which suggests that the pattern location learned is stable.
4. **Deployment Considerations**: The gap between public and private leaderboard scores suggests that further validation with a holdout set during the operational deployment phase may enhance generalization in a production environment.

An interesting conclusion from the above discussions is that our approach is a case for how machine-learning models, especially gradient boosting frameworks backed by solid feature engineering, can detect fraudulent transaction instances very accurately and thereby hold so much promise for real-life fraud prevention. Overall, our approach demonstrates that machine learning models, particularly gradient boosting frameworks with careful feature engineering, can effectively detect fraudulent transactions with high accuracy, offering significant potential for real-world fraud prevention systems.

# Discussion

The results obtained show that it is possible to apply machine learning for detection of fraud in the IEEE-CIS dataset successfully. Now, in this section, implications of such results are given, how one can introduce them to a real financial system, limitations and improvements, and ethical issues such as dealing with false positives and privacy.

## 5.1 Transaction Amount of Fraud

The analysis found that the propensity increases with tariff size, which naturally is assumed in the industry case. But there is a nuance to that. Financial institutions may want to apply a hard rule like "flag all transactions above $1000" - findings in this study show that it will not be missed by such a rule; it would catch a decent part of fraud (many tend to be over that threshold), yet it would also be capturing a lot of legitimate big purchases (vacations, electronics, etc.), which may not be acceptable.

It is much more sophisticated because it relates amounts to other signals. For example, a $50 transaction at an odd time with mismatched addresses might be riskier than a $500 transaction fitting a customer's normal pattern. That underscores why multivariate models are important over simple thresholds. Nevertheless, the knowledge that the propensity to fraud increases with amount may help in a strategy: banks could require progressively more stringent authentication for higher amounts (e.g. require SMS verification for purchases over a certain limit), this would be a kind of dynamically placed friction with behaviorally adhering to risks.

## 5.2 Model Performance and Feature Engineering Insights

The gradient boosting models that we have produce incredible results on the IEEE-CIS fraud detection dataset, with the best assembly model scoring gold medal. The best model ensemble combining LightGBM (80%) over CatBoost (20%) yielded diamond-like performance on the public leaderboard score at 0.963 and 0.942 for the private leaderboard. These results are a rational sign that our endeavors at featuring engineering and forays into model development were vast and extensive.

The variation between public and private leaderboard score (0.963 versus 0.942) indicates some overfitting as regards the public test set. This signifies the relevance of cross-validation in various dimensions and a careful tuning of hyperparameters. In a "real-world environment," this difference would mean that regular retraining and validation would be necessary to maintain model performance over time.

Our feature engineering approach focused on several key areas that proved critical to model success:

**Device and Identity Processing**

It is evident that extracting meaningful information from devices and identity fields provides solid signals for fraud detection. Device type, browser details, operating systems, etc., could be used to identify suspicious patterns, which take part in fraudulent activity. For example, the processing done by the id\_split () function of the columns DeviceInfo, id\_thirty, id\_31 would contribute immensely to the predictive power of the model.

**Temporal Feature Creation**

Transforming TransactionDT into more logical time units of days, weeks, months has resulted in the increased capturing of cyclical patterns of fraudulent activity in the model. These attributes DT\_D, DT\_W, and DT\_M manage to capture temporal patterns of fraud and are best helpful in production systems nowadays, with temporal fraud patterns arising very often around the clock.

**Transaction and Product Features**

The transaction counts and average fraud rates of every ProductCD category were calculated over time windows. Such features captured the patterns about a specific product and the temporal fraud trend associated with it (ProductCD\_W\_Day, ProductCD\_C\_Day, etc.).

**Unique Identifier Hashing**

Tracked patterns over different users and devices by means of features like such as uid1 (combination of card and address) and device\_hash. The ability of this technique could be perfected in a production environment to flag sophisticated fraud rings that could work across multiple accounts.

**Missing Value Strategy**

This allowed the model to learn patterns pertaining to missingness itself for the high degree of sparsity in the dataset with 414 features stacked with missing values. Dropping up information would not be the best possible approach in our case. But replacing missing entries with values beyond the normal range (e.g., -999) helps to learn patterns concerning missing entries from a given record.

**Categorical and Numerical Transformations**

Label encoded handling of categorical columns and various transformations count occurrences across categories, revise series D features, create interaction features, etc., increase the power for complex relationships captured by the model.

**Model Deployment Considerations**

The engineering steps involved in deploying an XGBoost or LightGBM model in a real-time scoring system are many. The model, being small, can score the event in milliseconds. This will fit nicely with real-time fraud checks which usually require execution in less than 100ms to avoid stalling transaction processing. The features that feed the model will also need to be computed in real-time. Some features such as "count of transactions for this card in last 24h" require event streaming or querying recent history; this can be achieved with a bit of heavy lifting around infrastructure, such as maintaining a rolling count in a database or cache. The identity features (e.g., device/browser info) assume that data is passed directly from the transaction (in an online purchase, the fingerprint is often collected by the merchant). Making sure that all this data is passed and standardized is a practical nightmare.

The second significant concern is data drift: it hinges on a fixed dataset which the model was trained on. Fraud tactics keep evolving, making it fundamental to implement monitoring for data drifting and model performance deterioration. A deployed model should undergo retraining with new data every so often. Performance metrics (especially recall – if it begins to drop, this may indicate that newly emerging fraud patterns are being missed) would be monitored by us. The alerts could also be supplemented with a feedback loop for retraining: when the investigators identify an alert as being false, or true fraud, this information can be harnessed to update the model.

**False positives vs. False negatives**

In our findings, we achieved commendable precision at the expense of recall. Whether that is the right balance depends on the business context. A false negative (one that is missing fraud) translates to straightforward financial losses and customer dissatisfaction (when customers notice unauthorized charges later). But a false positive (one that wrongly flagged a transaction) causes customer inconvenience (declines or verification calls), in addition to operational cost (manual reviews and support calls).

Reputational risk exists if a bank declines too many legitimate transactions; customers may become frustrated and switch institutions. Typically, banks err on the side of preventing false negatives up to a limit. They will not hesitate to annoy some customers to save big on losses from fraud; however, they try not to push it too far because annoyed customers are also loss-makers.

Our model's threshold can be tuned into based on business priorities:

1. **Implement a Tiered Approach**: For transactions with a high fraud possibility, an automated decline could suffice, while ones that are moderate could demand further verification or manual processing. For instance, auto-blocking with very high precision (firmly always right) on the first threshold is reasonable, while transactions above the second threshold (say probability >0.3, which might recall ~80% with precision ~50%) may simply be marked for manual review (allowed through, but alerting an investigation).
2. **Adjust Thresholds Based on Amount**: Higher value transactions can be exposed to stricter scrutiny; these recommendations follow from our analysis to show a relationship between transaction amount and fraud likelihood.
3. **Supplement with Rule-Based Systems**: Although a machine learning model is in place, banks keep some expert rules for regulatory compliance (certain transactions might legally need flagging, like those on sanction lists) and domain knowledge (e.g. "if 10 failed attempts on card then success, likely fraud" might not be an obvious pattern in static historical data but is an old-timer).

**Explainability and Trust**

The key to the practical application of ML in fraud detection is communication. Feature importance analysis from our models sheds light on the path decisions take. In practice, one could pre-compute some explanation logic so that, whenever a transaction is flagged, the system could generate a reason code such as: "Flagged due to high-risk profile: large amount, unusual device, and mismatched address compared to previous transactions."

Many banks require such reason codes to be retained for each alert, to help in the audit trail and quickly orient investigators to the nature of the alert. Our feature importance findings could be summarized into a common set of reason templates. Sometimes, though, a model can find combinations of factors that are harder to explain simply ("it just doesn't fit the usual pattern"), where advanced interpretability or a human analyst review are needed.

Regulators are putting increased pressure on AI systems, especially in banking, to provide explainability (the EU's GDPR has included a "right to explanation" for algorithmic decisions). Therefore, our mapping for the explanation based on feature importance and existing knowledge of the features would work perfectly to meet these requirements.

**Ethical and Fairness Considerations**

A fraud detection model might bring in biases if certain demographic groups generate diverse types of transactions. In this case, the model flags disproportionately transactions coming from one or another demographic group. Our data set is anonymized. However, consider that "addr2" (country) or specific email domains might be correlated with protected characteristics like nationality. We should check that the model is not unfairly targeting a group.

Card transactions make it a little less morally clear than, say, loan approvals: one thing is catching the criminals with credit card fraud. But, for instance, stealing learning that transactions from a certain region (attribute-wise, a region with one ethnicity) are riskier and goes on to discriminate that way. It may be mitigated by testing on segments to ensure that it is not unnecessarily harsher on one group of legitimate users.

Our model includes identity features (device, IP, email) – using, which is important for fraud detection, but these features infringe upon privacy. Granted, users consent to fraud monitoring as part of their banking but should be treated with sensitivity and kept in secure data stores in our model. Federated learning would allow us to deploy that model privacy-preservingly if several banks collaborated. Otherwise, in a single entity, data is stored and used only within the bank.

**Model Limitations and Future Improvements**

While our performance is strong, several limitations and opportunities for improvement exist:

1. **Dataset Timeframe**: The IEEE-CIS dataset corresponds to a specific period that may, therefore, hinder the model from generalizing to fraud patterns that may be relevant now. This "seasonal" pattern of shopping may not be directly modeled into the pattern, so the model may become overfit about some artifacts of the dataset.
2. **Fraud Type Specificity**: This is a transactional fraud detection - different than credit card application fraud or identity theft detection, so we cannot generalize to other types.
3. **Feature Engineering Extensions**: Further performance improvements are possible from additional, such as:
   * **Deep learning with sequence data**: time-linking transactions by user (RNNs) to check whether the improvement in catching spree fraud is brought by sequence modeling.
   * **Graph link analysis**: Connect entities such as cards, devices, and IPs in a graph and find coordinated fraud rings by utilizing graph neural networks or community detection.
   * Using an autoencoder for anomaly detection, by scoring outliers as potential fraud after training an autoencoder only on genuine transactions, then comparing or ensembling with our supervised model.
4. **Transfer Learning**: It would be interesting to see whether our model features and importance hold on to other data (e.g., a public credit card dataset from a different bank) to gauge generalizability.
5. **Cost-sensitive Optimization**: This should incorporate AUC and a cost matrix where false negative cost = average fraud amount lost and false positive cost = operational cost to optimize for business outcomes.
6. **Continuous Learning**: An adaptive learning solution which learns in an incremental way as soon as labeled data arrives should ensure the system is kept up to date against the ever-changing tactics of fraud.

**Collaboration Between Model and Analysts**

In the fraud department of the bank, fraud models serve as an assistant to the able analysts. Their job would involve reducing the caseload by eliminating noise and prioritizing fraud. The high precision of our model means that the analyst spends extraordinarily little time on false cases. It is a huge gain in efficiency historically; simple systems would generate more false alarms than true ones, weighing upon the analysts.

Better precision translates into an improvement in workflow and employee morale (analysts focus on real fraud). Otherwise, if recall is poor, whatever fraud is left will be encountered by the analysts or another system (often customer reporting or other second-level checks). Ideally, therefore, as the models continue to mature, there should become a reduction in the reliance on customers notifying about fraud (less "reactive" detection).

# Conclusion

This project has successfully applied machine learning in detecting patterns of financial transactions classified as frauds using the IEEE-CIS dataset; insights were given to the role of transaction amount on fraud risk. Gradient-boosted trees were used to train a high-performance fraud classifier (AUC ≈ 0.95) capable of detecting over one-half of fraudulent transactions with very few false alarms. Deep exploration analysis and statistical tests were used to show that higher amounts were associated more often with fraud transactions than with genuine ones, such as during the late-night hours. However, it was also shown that no one factor is adequate; many signals must be combined for effective fraud detection.

**Key findings and contributions:**

* We confirmed that the distribution of transaction amounts is significantly different for fraud versus non-fraud transactions (p < 0.001 via Mann-Whitney U), supporting the intuition that higher amounts entail greater fraud risk. For instance, median fraud amounts were double median legit amounts in this dataset. This answers our research question affirmatively, but with the caveat of the overlapping distributions indicating amount alone is not a definitive predictor.
* A machine learning pipeline has been developed for the processing of an exceptionally large and complex dataset with 400+ features: data merging, missing value imputation, categorical data encoding, and feature engineering (time-of-day, frequency counts, address match flags). These preprocessing steps (explained in full in Appendix A) can be reused for similar fraud-related datasets.
* Among many tested models, the ensemble tree models (XGBoost/LightGBM) had the best performance. The final model (XGBoost) achieved a precision of ~94% and recall of ~53% on the validation set, with an F1-score of 0.68 for the fraud class. This significantly outvalues the baseline methods and simpler models themselves (for example, logistic regression had F1 ~0.28). The model's high precision indicates a significant reduction in false-positive alerts in an operational environment.
* We could add interpretability to the "black box" model through SHAP explanations. We established which features were most influential (e.g., transaction amount, card usage frequency, device/browser info, address/email mismatch, and a few anonymous features). We demonstrated how some feature values (such as an exceptionally large amount or a rare device type) increase or decrease the fraud risk score. This is especially important for the trustworthiness of stakeholders and for compliance; it also produced some intuitive rules-of-thumb (e.g., unusual spending patterns or technical fingerprint-raise flags).
* We then describe how to operationalize and optimize the model: threshold tuning for false positives vs false negatives and integration into existing fraud prevention frameworks. This model can be personalized according to risk appetite and combined with rules-based strategies.
* In fact, ethics addressed the minimum impact on legitimate customers, which the model already has since it increased precision and decreased false declines. Equity and continuous monitoring were also issues considered to ensure that the model's decision-making is unbiased and remains effective with the changing nature of fraud.

**Recommendations:**

For financial institutions or practitioners willing to spend for similar applications

1. **Use multi-factor models.**

Our finding confirms that relying on one attribute (such as amount or location) does not suffice. For the complex nature of fraud data, ensemble methods capable of handling dozens of features are recommended.

1. **Address class imbalance.**

stratified sampling, proper metrics (AUC, precision/recall), employ some form of oversampling (like SMOTE/ADASYN) should be in the toolkit. We have found parameter tuning (e.g., scale\_pos\_weight) for boosting algorithms to manage the imbalance impressively but must monitor recall.

1. **Incorporate explainability**

It is hard to get an opaque model accepted without any sort of explanation. We have shown how such insight is gained with tools like SHAP; teams should start integrating and generating human-readable reason codes for every alert.

1. **Regularly retrain and validate**

The patterns of fraud change, and thus a model's performance will drop once it is static for a while. We recommend retraining with fresh data now and again (monthly or quarterly) and confirming that the feature importance does not hugely differ, or the error rates are not drastic. Input distribution anomalies may even include a new type of payment method or device and should trigger updates on the model.

1. **Hybrid approach**

A hybrid approach could combine the strengths of data-oriented modeling and expert judgment. For instance, it could keep the essential rule sets in place (to cover situations the model can or could not easily model) while allowing the ML model to take care of most cases. This work could, therefore, further rule-based systems in enhancing nuance in the so-called 'gray area' decisions.

Thus, the project has once again turned out successfully because its objectives had been met within the completed project analysis which blends domain intuition with statistical as well as machine learning capability. This is indeed an end state fraud detection model along with validations by the insight into the amounts involved in the transaction. However, these insights have been translated into practical recommendations and considerations for their deployment. Statistical approaches alongside machine learning widen the realm of understanding fraud dynamics and viable solutions toward mitigating fraud risks. That is why technology needs to be continuously researched and made to re-adapt with fraudsters evolving. This report and model are indeed the steppingstone for developing more secure and intelligent financial transaction systems that will safeguard businesses and consumers against the omnipresent threat of fraud.

# References

1. A. Abdallah, M. A. Maarof, and A. Zainal, “Fraud detection system: A survey,” *Journal of Network and Computer Applications*, vol. 68, pp. 90–113, 2016.
2. J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, “Credit card fraud detection using machine learning techniques: A comparative analysis,” in *Proc. Intl. Conf. Computing Networking and Informatics (ICCNI)*, 2017, pp. 1–9.
3. A. C. Bahnsen, D. Aouada, A. Stojanovic, and B. Ottersten, “Feature engineering strategies for credit card fraud detection,” *Expert Systems with Applications*, vol. 51, pp. 134–142, 2016.
4. S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, “Data mining for credit card fraud: A comparative study,” *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.
5. R. J. Bolton and D. J. Hand, “Statistical fraud detection: A review,” *Statistical Science*, vol. 17, no. 3, pp. 235–255, 2002.
6. N. Carneiro, G. Figueira, and M. Costa, “A data mining-based system for credit-card fraud detection in e-tail,” *Decision Support Systems*, vol. 95, pp. 91–101, 2017.
7. M. Carminati, R. Caron, F. Maggi, I. Epifani, and S. Zanero, “BankSealer: A decision support system for online banking fraud analysis and investigation,” *Computers & Security*, vol. 53, pp. 175–186, 2015.
8. 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 Systems with Applications*, vol. 41, no. 10, pp. 4915–4928, 2014.
9. J. Jurgovsky et al., “Sequence classification for credit-card fraud detection,” *Expert Systems with Applications*, vol. 100, pp. 234–245, 2018.
10. R. A. Leite et al., “EVA: Visual analytics to identify fraudulent events,” *IEEE Trans. Visualization and Computer Graphics*, vol. 24, no. 1, pp. 330–339, 2018.
11. 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,” *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.
12. R. Patidar and L. Sharma, “Credit card fraud detection using neural network,” *International Journal of Soft Computing and Engineering*, vol. 1, pp. 32–38, 2011.
13. PwC, **“Global Economic Crime and Fraud Survey 2022.”** PricewaterhouseCoopers, 2022. [Online]. Available: *https://www.pwc.com/fraudsurvey2022*
14. V. Van Vlasselaer et al., “APATE: A novel approach for automated credit card transaction fraud detection using network-based extensions,” *Decision Support Systems*, vol. 75, pp. 38–48, 2015.
15. J. West and M. Bhattacharya, “Intelligent financial fraud detection: A comprehensive review,” *Computers & Security*, vol. 57, pp. 47–66, 2016.
16. K. G. Al-Hashedi and P. Magalingam, “Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019,” *Computer Science Review*, vol. 40, art. 100402, 2021.
17. U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, “Using generative adversarial networks for improving classification effectiveness in credit card fraud detection,” *Information Sciences*, vol. 479, pp. 448–455, 2019.
18. S. Makki et al., “An experimental study with imbalanced classification approaches for credit card fraud detection,” *IEEE Access*, vol. 7, pp. 93010–93022, 2019.
19. A. Pumsirirat and L. Yan, “Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, pp. 18–25, 2018.
20. D. Varmedja, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, “Credit card fraud detection – machine learning methods,” in *Proc. 18th International Symposium INFOTEH-JAHORINA*, 2019, pp. 1–5.
21. I. Y. Hafez et al., “A systematic review of AI-enhanced techniques in credit card fraud detection,” *Journal of Big Data*, vol. 12, no. 6, 2025.
22. F. Alarfaj and S. Shahzadi, “Enhancing fraud detection in banking with deep learning: Graph neural networks and autoencoders for real-time credit card fraud prevention,” *IEEE Access*, vol. PP, 2024 (Early Access).
23. **IEEE-CIS Fraud Detection Dataset**, Vesta Corporation, 2019. [Online]. Available: *https://www.kaggle.com/c/ieee-fraud-detection*
24. S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Proc. 31st NIPS*, 2017, pp. 4765–4774.
25. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
26. G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in *Proc. 31st NIPS (Workshop on ML Systems)*, 2017.
27. A. Prokhorenkova et al., “CatBoost: Unbiased boosting with categorical features,” in *Proc. 32nd NeurIPS*, 2018, pp. 6639–6649.
28. J. Boitnott, **“Credit Card Fraud Statistics (2023).”** Self-Financial, Jan. 2024. [Online]. Available: *https://www.self.inc/info/credit-card-fraud-statistics/*
29. Y. Tang and Y. Liang, “Credit card fraud detection based on federated graph learning,” *Expert Systems with Applications*, vol. 213, art. 118902, 2024.
30. Stripe, **“Fraud detection using machine learning: What to know.”** [Online]. Available: [*https://stripe.com/blog/machine-learning-fraud-detection*](https://stripe.com/blog/machine-learning-fraud-detection)

# Appendices

**Appendix A: Hyperparameters and Training Settings**

This appendix provides the hyperparameters used for the final models after tuning, as well as any relevant training settings:

* **XGBoost (Final Model):**
  + max\_depth = 6
  + learning\_rate = 0.05 (eta)
  + n\_estimators = 500 (with early stopping, the model stopped at ~320 trees)
  + subsample = 0.8 (80% of data sampled per tree)
  + colsample\_bytree = 0.4 (40% of features sampled per tree)
  + min\_child\_weight = 10 (to make trees slightly less complex, handling noise)
  + gamma = 0 (no minimum loss reduction for split, as overfitting was not severe with other params)
  + scale\_pos\_weight ≈ 27 (this was set to inverse of fraud ratio = 96.5/3.5)
  + Training rounds used early stopping on eval AUC with early\_stopping\_rounds = 50.
  + The best iteration had AUC 0.953 on validation. Training time ~15 minutes on CPU.
* **LightGBM:**
  + num\_leaves = 63 (equivalent to max\_depth ~ 6 or 7 for Light; LightGBM can manage more leaves due to leaf-wise growth)
  + learning\_rate = 0.05
  + n\_estimators = 500 (stopped around 400 trees early stop)
  + feature\_fraction = 0.4 (like colsample)
  + bagging\_fraction = 0.8 and bagging\_freq = 1 (subsample 80% each iteration)
  + min\_data\_in\_leaf = 50
  + scale\_pos\_weight ≈ 27 (or equivalently used class\_weight in API)
  + LightGBM achieved AUC ~0.950 on validation, remarkably close to XGBoost. Time ~10 minutes.
* **CatBoost:**
  + depth = 6
  + learning\_rate = 0.1 (CatBoost can sometimes use higher lr and still converge fast)
  + iterations = 1000 (early stopped around 600)
  + loss\_function = Logloss with eval\_metric = AUC
  + od\_wait = 50 (early stop rounds)
  + random\_strength = 1.0 (CatBoost specific regularization)
  + scale\_pos\_weight = 27 (class weight for imbalance)
  + We specified categorical features: CatBoost automatically managed encoding for features like card4, ProductCD, etc.
  + CatBoost AUC ~0.941. Training ~30 minutes (CPU).
* **Neural Network (MLP):**
  + Architecture: two hidden layers, sizes [100, 50] (we tried [64,32] as well, but 100/50 gave a slight edge).
  + Activation: ReLU for hidden, Sigmoid for output.
  + Optimizer: Adam with learning\_rate = 0.001.
  + Loss: Binary crossentropy.
  + Batch size: 1024. Epochs: trained for 10 epochs (early stopped on val loss improvement ceasing).
  + Class weights: 0:1.0, 1:27.0} to manage imbalance.
  + We also tried dropout regularization (0.2) on hidden layers which helped generalization slightly.
  + Final NN AUC ~0.90. Precision ~0.30, recall ~0.55 at threshold 0.5.
  + Training time ~1 hour (CPU) or a few minutes on GPU.
* **Logistic Regression (for baseline):**
  + We used an L2-regularized logistic regression via scikit-learn.
  + Regularization strength C = 0.1 (to prevent overfitting given many features).
  + We limited the features to about fifty top features (used univariate selection on training) because using all four hundred led to worse performance due to multicollinearity and noise.
  + Even with these, logistic reg had AUC ~0.88.

These hyperparameters were chosen after trying a range: for example, XGBoost max\_depth eight did not improve AUC but increased overfit; lower learning rates (0.01) required more trees and did not yield much better AUC, etc. The given settings balanced bias-variance well.

**Threshold selection:** While metrics were often reported at 0.5 threshold, in practice we would choose a threshold based on precision-recall curve. For the XGBoost model, we considered the threshold that gives precision ~0.95 as our operating point (~0.5 in this case). If needed, one could formally choose threshold by maximizing F1 or a weighted utility. We note that 0.5 is not inherently special since the classes are imbalanced; it just so happened that 0.5 was near optimal for F1 in our case after using scale\_pos\_weight.

**PROJECT PROCESS DOCUMENTATION TEMPLATE**

**Student: Shazib Armaan Patras Supervisor: Martins Olaleye**

**Meeting Number: Date/Time: 21/04/2025**

**Agenda for meeting:** (*Should be set and sent to the supervisor before the meeting*)

**Discussion of agenda items:**

**Summary of agreed action plan:**

**Notes:**

University of the West of Scotland

School of Computing, Engineering and Physical Sciences

MSc IT with Data Analytics

Student name: Shazib Armaan Patras

Banner ID: B01750852

Email: B01750852@studentmail.uws.ac.uk

Project being undertaken on a part-time or full-time basis: Full time

MSc Programme (specify the specialist pathway, if any): MSc IT with Data Analytics

MSc Programme Leader: Martins

Project Title:

|  |
| --- |
| ***Fraud Detection in Financial Transactions: A Machine Learning Approach Using IEEE-CIS Dataset*** |

**Research Question to be answered:**

|  |
| --- |
| ***How can machine learning algorithms be utilized to predict fraudulent financial transactions and improve fraud detection systems in the financial industry?*** |

**Overview, Justification and overall aim of project:**

|  |
| --- |
| **Overview**  Fraudulent financial transactions are a major concern for financial institutions worldwide, leading to billions in losses annually. Conventional approaches to fraud detection based on manual investigation and rule systems are fast dwindling in effectiveness through high false positives and limited capacity to detect complex fraud patterns. A much more scalable and effective solution comes in the form of machine learning, which can pick patterns hidden in transaction data, enabling financial institutions to detect and avert fraud in real-time, preventing monetary losses as they occur.  **Justification**  **Project Background and History of the Problem:**  The global financial sector is fighting fraud all the time by bringing into question outdated systems of detecting fraud as fraudsters continue to have experiences with existing rules. Machine learning offers a data-driven approach wherein patterns could be learned automatically from historical transaction data so the detection can be improved. However, this is accompanied by issues concerning class imbalance, feature engineering, and model interpretability, which is where this project would investigate even more concerning how ML could further improve the detection of fraud on financial transactions.  **What is Available**   * Traditional fraud detection methods (e.g., rule-based systems, manual reviews). * Machine learning approaches, such as ensemble methods (e.g., XGBoost, LightGBM), and neural networks. * Existing datasets, such as the IEEE-CIS Fraud Detection dataset, which contains transaction data labeled as fraudulent or non-fraudulent.   **What is Missing or Needs Improvement**   * Existing models often struggle with imbalanced data, where fraudulent transactions make up less than 5% of the total. * Models need to be interpretable for stakeholders such as regulators and financial analysts. * Real-time fraud detection systems need to be both fast and accurate, minimizing false positives without missing fraud cases.   **Why is It Worth Working On**  This project is so important since it targets an emerging need for the financial sector, with much more robust, real-time, and accurate fraud detection systems being demanded. The huge volume of transaction data that machine learning can analyze warrants that the reduction of fraud cases results in saved financial amounts and improved customer experience by ultimately minimizing false alarms.  **Justification**  **This project is essential as it aims to:**  This research attempts to create a machine learning framework to predict fraudulent financial transactions using advanced algorithms. The project aims to:   * Build models that accurately identify fraudulent transactions. * Minimize false positives, thus reducing unnecessary manual reviews and enhancing customer experience. * Provide a scalable and adaptable fraud detection system. * Use techniques such as SHAP for model interpretability, ensuring transparency for stakeholders.   **Rationale for the Research:**  Although fraud detection has seen tremendous advancements, the rule-based systems still in use by many organizations date back to previous decades. The results of these systems can hardly match those of today's growing fraud. Machine learning provides an environment for continual learning and adaptation, thereby spotting patterns in data that standard systems would otherwise overlook. The project intends to make further advances in fraud detection using state-of-the-art ML techniques for improved accuracy and efficiency.  **Key Gaps Identified:**   * Current fraud detection systems fail to adapt to evolving fraud tactics. * Many ML models produce high false positives. * A lack of transparency in how models make decisions, leading to a lack of trust in their predictions. |

**Objectives:**

|  |
| --- |
| The primary goal of this research is to predict fraudulent financial transactions using machine learning algorithms. The specific objectives are:   * Gather a comprehensive dataset (e.g., IEEE-CIS Fraud Detection dataset) containing transaction and identity data. * Analyze transaction data to identify key features, such as transaction amount, time of day, device information, and address mismatches. * Develop and evaluate machine learning models, including XGBoost, LightGBM, and neural networks, to predict fraud. * Compare the performance of these models to determine the best approach for fraud detection. * Provide actionable recommendations for improving fraud detection systems in real-world financial institutions. |

**Methodology:**

|  |
| --- |
| This research will follow a structured approach to predict fraud in financial transactions using machine learning techniques:   * **Dataset Collection**   The dataset used will be the IEEE-CIS Fraud Detection dataset, which contains transaction records labeled as fraudulent or non-fraudulent.   * **Data Preprocessing**   Raw data has many discrepancies that can distort the machine learning models. Here preprocessing includes:   * Cleaning and validating the data to remove redundant or irrelevant features. * Handling missing values, especially in identity-related features. * Encoding categorical features and normalizing continuous variables such as transaction amounts. * Addressing class imbalance using techniques like SMOTE or adjusting the model's class weights.   **Feature Engineering**   * Extracting relevant features such as time of day, transaction amount, card details, and address information. * Using SHAP (Shapley Additive exPlanations) to interpret model predictions and identify the most influential features.   **Model Training**  The following machine learning algorithms will be used:   * **XGBoost:** A gradient-boosting method known for its effectiveness in tabular data and handling class imbalance. * **LightGBM:** Another gradient-boosting method that handles large-scale datasets and categorical features efficiently. * **Neural Networks:** A simple multi-layer perception to compare performance with tree-based models.   **Evaluation and Validation**  Models will be evaluated based on metrics such as ROC AUC, precision, recall, and F1 score. Cross-validation will be used to ensure robust evaluation of model performance. |

**Work Plan:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activity | Year 2025 | | | | | | | | | | | |
| Months | | | | | | | | | | | |
| Feb | | | | March | | | | April | | | |
| 1st week | 2nd week | 3rd week | 4th week | 1st week | 2nd week | 3rd week | 4th week | 1st week | 2nd week | 3rd week | 4th week |
| **Proposal** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Objective and Scope** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Literature Review** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Research Methodology** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Collect Data** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Data Assessment** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Technology Performance** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Results Overview and Discussion** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Summary and Suggestions** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Crafting a Compelling Thesis** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Refining Project** |  |  |  |  |  |  |  |  |  |  |  |  |
| **Final Submission** |  |  |  |  |  |  |  |  |  |  |  |  |

**Relationship of proposed project to MSc programme/stream:**

|  |
| --- |
| The project entitled 'Detecting Fraudulent Financial Transactions using Machine Learning Algorithms' is an exciting undertaking that complements both the theoretical and applied areas of study of the MSc program. Fraud detection in finance is a very important area with considerable potential for effective machine learning applications because detection will happen before the financial loss occurs due to a transaction. Following is a summary of how the research links to key points covered under the MSc program:  **Connection to Financial Analytics**  The project addresses a major challenge in the financial industry, fraudulent transactions. The research thus uses machine learning to assist decision-makers at financial institutions with preliminary solutions for detecting fraud and for improving their detection systems. This supports the MSc theme of an analytical technique applied to a complex business problem.  **Connection to Machine Learning and Artificial Intelligence**  Examples of such models include theories behind other machine learning models like XGBoost, LightGBM, or even neural networks to practically demonstrate artificial intelligence in fraud detection. This methodology applies various AI approaches to recognize patterns from huge financial datasets, moving beyond popularly utilized reactive fraud detection models into proactive and predictive modeling. Thus, this shows the importance of machine learning about practical problems in the financial sector: one of the focuses of the MSc program.  **Connection to Data Science and Big Data**  The project thus applies different data science principles such as feature engineering, preprocessing of data, and exploration data analysis (EDA). These represent the essence of what it takes to handle big data and pick definite patterns from very huge data sets, making a case for conducting experiments on transaction data on a large scale. The study here thus bridges the divide between theory and practice regarding big data in order that actions may be taken- one of the main thrusts of the MSc.    **Connection to Business and Organizational Strategy**  Fraudulent transactional activities directly affect a corporation's health and performance. This connects business technology strategy to actionable insights and organizations toward improving their fraud detection systems. It shows how advanced machine learning techniques can empower financial institutions to base effective decision-making on data and thereby improve an organization's efficiency and security.  **Connection to Industry and Practical Applications**  The results of this research would create a high impact on the industries prone to fraud detection, including banking, insurance, and e-commerce. This work provides not only theoretical but also practical evidence for the implementation of machine learning in addition to systems for fraud detection in real time. A direct reflection of how the MSc engenders students for solution applications across various industry contexts includes applicability across different business sectors.   * **Academic and Research Contributions**   The application of machine learning techniques in fraud detection has been further enhanced by this project, thus contributing to academic research. It helps to fill gaps within the literature about the practical implementation of machine learning models in the financial services sector. Moreover, the project elaborates on model interpretability, employing methods such as SHAP, which can be applicable in situations where machine learning models require transparency and trust.  The project further shows the interdisciplinary dynamics existent in the MSc program by applying data science, machine learning, and business strategies to solve current problems in finance. The application of theory and practice in the project shows the program's preparedness of students to tackle truly complicated challenges that happen to exist in any industry. |

**Indicative reading list (references to be correctly presented):**

|  |
| --- |
| Liu, Y., & Zhuang, H. (2023). Machine Learning in Fraud Detection: A Comparative Analysis. *Journal of Financial Technology*, 12(4), 112-124.  Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.  Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 3146–3154.  Lundberg, S. M., & Lee, S. I. (2017). A Unified Approach to Interpreting Model Predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4765–4774.  Fallucchi, F., Coladangelo, M., Giuliano, R., & De Luca, E. W. (2020). Predicting employee attrition using machine learning techniques. Computers, 9(4), 1–17. https://doi.org/10.3390/computers9040086  Geiler, L., Affeldt, S., & Nadif, M. (2022). A survey on machine learning methods for churn prediction. International Journal of Data Science and Analytics, 14(3), 217–242. https://doi.org/10.1007/s41060-022-00312-5  Hom, P. W., Mitchell, T. R., Lee, T. W., & Griffeth, R. W. (2012). Reviewing employee turnover: Focusing on proximal withdrawal states and an expanded criterion. Psychological Bulletin, 138(5), 831–858. https://doi.org/10.1037/a0027983  Kaushal, N., Kaurav, R. P. S., Sivathanu, B., & Kaushik, N. (2023). Artificial intelligence and HRM: identifying future research Agenda using systematic literature review and bibliometric analysis. Management Review Quarterly, 73(2), 455–493. https://doi.org/10.1007/s11301-021-00249-2  Kiran, P. R., Chaubey, A., & Shastri, R. K. (2024). Role of HR analytics and attrition on organisational performance: a literature review leveraging the SCM-TBFO framework. Benchmarking: An International Journal, 31(9), 3102–3129. https://doi.org/10.1108/BIJ-06-2023-0412 |

**Marking scheme**

|  |
| --- |
|  |

**Supervisor**

|  |
| --- |
| Martins Olaleye |

**Moderator**

|  |
| --- |
| Ashraf Mahmud |

**Programme Leader**

|  |
| --- |
| Martins |

**Date specification submitted**

|  |
| --- |
| 03/03/2025 |

Please complete the ‘ethics’ & pathway confirmation form below for all projects.

**School of Computing, Engineering and Physical Sciences**

**MSc IT with Data Analytics– REQUIREMENT FOR ETHICAL APPROVAL & PATHWAY CONFIRMATION**

**SECTION 1: TO BE COMPLETED BY THE STUDENT**

Does your proposed research involve: research with human subjects (including requirements gathering and product/software testing), access to company documents/records, questionnaires, surveys, focus groups and/or other interview techniques? Does your research entail any process which requires ethical approval? (please enter √ in the appropriate box)

|  |  |  |
| --- | --- | --- |
| YES |  | **You must apply for approval to the Ethics Review Manager** |
| NO |  | You do not need to apply to the Ethics Review Manager |

I confirm that the above specification aligns with my MSc programme specialist pathway. (please enter √ in the box)

**Name of Student (Print name): Shazib Armaan Patras**

**Signature: SHAZIB**

**Date: 21/04/2025**

**SECTION 2: TO BE COMPLETED BY THE PROJECT SUPERVISOR**

I understand that the above project requires/does not require\* ethical approval (\*please delete as appropriate).

I confirm that the above project aligns with the MSc programme specialist pathway the

student is enrolled in. (please enter √ in the box)

**Supervisor (print name):**

**Signature**:

**Date:**