DISSERTATION

DETECTION OF FRAUDULENT CUSTOMER TRANSACTIONS

Candidate number:

277195

Supervisor:

Prof. Miroslav Chlebik

Department:

School of Mathematics & Physical Sciences

Course:

MSc Data Science

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor, Dr. Miroslav Chlebik, from the Department of Mathematics, for his exceptional guidance and unwavering support throughout the course of this dissertation. His profound knowledge, insightful feedback, and continuous encouragement have been invaluable in shaping this project. I am truly thankful for the time and effort he has dedicated to helping me navigate the complexities of this research, and for fostering an environment where I could thrive intellectually.

I am also immensely grateful to my mentors, Adam Barrett and Dr. Dhruva Raman, whose expertise in data science has been instrumental in the development of this dissertation. Their thoughtful advice, constructive criticism, and willingness to engage with my ideas have greatly enriched my understanding and have been a driving force behind the successful completion of this work.

I would also like to extend my sincere thanks to the University of Sussex, particularly the labs and the teaching assistants who provided invaluable support during my lab sessions. The hands-on experience and assistance I received in the labs were crucial in applying theoretical concepts to practical problems, and I am deeply appreciative of the learning environment they helped create.

Lastly, I want to extend my deepest gratitude to my family and friends for their steadfast encouragement and support throughout this journey. Their faith in my abilities has been a constant source of inspiration, and I am eternally thankful for their presence in my life.

ABSTRACT

Fraud detection and prevention are paramount challenges in both the financial and retail industries, impacting organizational integrity and profitability. As e-tailing becomes increasingly popular, sophisticated fraud methods, particularly in product returns, present significant risks to retailers. Flexible return policies, crucial for customer satisfaction, often lead to increased instances of return fraud, necessitating a delicate balance between maintaining consumer trust and protecting profits.

In the financial sector, the rise in sophisticated fraudulent activities has outpaced traditional rule-based detection methods, underscoring the need for advanced and scalable fraud detection systems. This dissertation explores the development and application of cutting-edge machine learning algorithms and data analytics to identify fraudulent transactions accurately while reducing false positives. The focus is on both supervised and unsupervised learning models, such as support vector machine (SVM), k-nearest neighbour (KNN), isolation forest, and local outlier factor (LOF). Ensemble modeling techniques are also examined for their ability to integrate multiple algorithms, thereby enhancing detection accuracy.

Effective fraud detection requires the analysis of huge volumes of transaction data in near-real-time or real-time to uncover suspicious patterns or anomalies. This dissertation highlights proactive methods, including dynamic risk scoring and adaptive thresholds, which are essential for real-time fraud prevention. Additionally, managing the massive volumes of financial data necessary for successful fraud detection relies on effective data handling techniques like distributed storage, real-time processing, and in-memory processing.

Key challenges addressed include handling imbalanced datasets, ensuring the privacy and security of sensitive data, and maintaining low latency to prevent transaction processing delays. By leveraging advanced technologies such as machine learning and artificial intelligence, this research aims to develop robust fraud detection systems capable of evolving with new fraud types, ensuring continuous protection against financial and retail fraud.

This study aims to contribute significantly to the field by providing a comprehensive framework for implementing advanced fraud detection models, ultimately safeguarding both financial institutions and retail businesses from the ever-evolving threat of fraud.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

INTRODUCTION

Fraud has always been a serious concern in financial transactions, with the potential to cause substantial financial losses and damage to reputations. As digital technologies have advanced, so too have the methods employed by fraudsters. The shift towards e-commerce and digital banking has introduced unprecedented efficiency and convenience for consumers but has also created new opportunities for fraudulent activities. The increasing reliance on online transactions, coupled with the growing complexity and volume of data, has made it essential to develop more sophisticated fraud detection systems. Traditional approaches, often dependent on predefined rules and heuristics, are becoming increasingly inadequate against the backdrop of evolving fraud strategies. As a result, there has been a growing interest in leveraging machine learning and artificial intelligence to enhance the detection and prevention of fraudulent transactions. This dissertation investigates the application of machine learning techniques in the detection of fraudulent transactions, with a particular focus on the opportunities and challenges presented by these advanced methods in the context of e-commerce and digital banking.

1.1 Background: Overview of Fraud in Financial Transactions and Its Implications

Fraud in financial transactions has been a persistent issue, threatening the integrity of financial systems and the trust of consumers. The traditional forms of fraud, such as check fraud and identity theft, have been compounded by the rise of digital transactions, which offer fraudsters new avenues for exploitation. The implications of fraud are far-reaching, affecting not only the victims but also the institutions responsible for safeguarding financial transactions. Financial losses due to fraud can be enormous, and reputational damage to institutions can result in a loss of customer trust, which is difficult to recover. The evolution of fraud has necessitated a parallel evolution in the methods used to detect and prevent it, making it a critical area of research and development. (Tanant, 2018)

Problem Statement: The Need for Effective Fraud Detection Methods

As fraudulent activities, particularly in the digital space, have grown more sophisticated, they have outpaced traditional fraud detection methods that often depend on static rules and manual oversight. These methods are increasingly inadequate in detecting new and evolving forms of fraud, leading to significant financial losses and breaches of consumer

trust. The need for more dynamic and adaptive fraud detection systems is evident, as fraudsters continuously develop new techniques to circumvent existing controls. Machine learning, with its ability to identify complex patterns and analyze large datasets, offers a promising solution to this problem. However, implementing these systems presents challenges, including managing imbalanced datasets, adapting to concept drift, and minimizing false positives while maintaining high detection accuracy.

Research Objectives: What Our Study Aims to Achieve

This study aims to explore the application of machine learning techniques in the detection of fraudulent transactions within the context of digital banking and e-commerce. The primary objectives are to:

1. Evaluate the effectiveness of various supervised and unsupervised machine learning models, such as Support Vector Machines (SVM), Random Forests, Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks, in identifying fraudulent activities.

2. Investigate the challenges associated with the implementation of these models, particularly in terms of handling imbalanced datasets and adapting to changing fraud patterns.

3. Propose a comprehensive framework for the integration of machine learning models into existing fraud detection systems, aiming to improve their accuracy and robustness.

Significance of the Study: Importance of the Research in the Field of Fraud Detection

This study's significance lies in its potential to contribute to the development of more effective and efficient fraud detection systems. By leveraging advanced machine learning techniques, this research addresses critical gaps in current fraud detection methodologies, particularly in handling the complexities of digital transactions. The results of this study could aid financial institutions and e-commerce platforms in better safeguarding their customers and mitigating the financial impact of fraud, thereby strengthening the overall security of financial systems.

1.2 Credit Card Fraud Detection System

Credit card fraud detection systems are designed to analyze transaction records and identify potentially fraudulent activities. These systems typically rely on a variety of features

such as transaction amounts, timestamps, merchant information, and cardholder details to distinguish between legitimate and fraudulent transactions. As digital transactions have become more prevalent, the complexity and volume of data processed by these systems have increased significantly. To manage this, credit card fraud detection systems employ both expert-driven and data-driven approaches. In an expert-driven system, rules and heuristics are developed by domain experts based on their knowledge of fraudulent behavior. These rules might include thresholds for transaction amounts or patterns of spending that deviate from the norm. However, as fraudsters develop more sophisticated methods, these traditional rule-based systems often struggle to keep pace. (Alemad, n.d.)

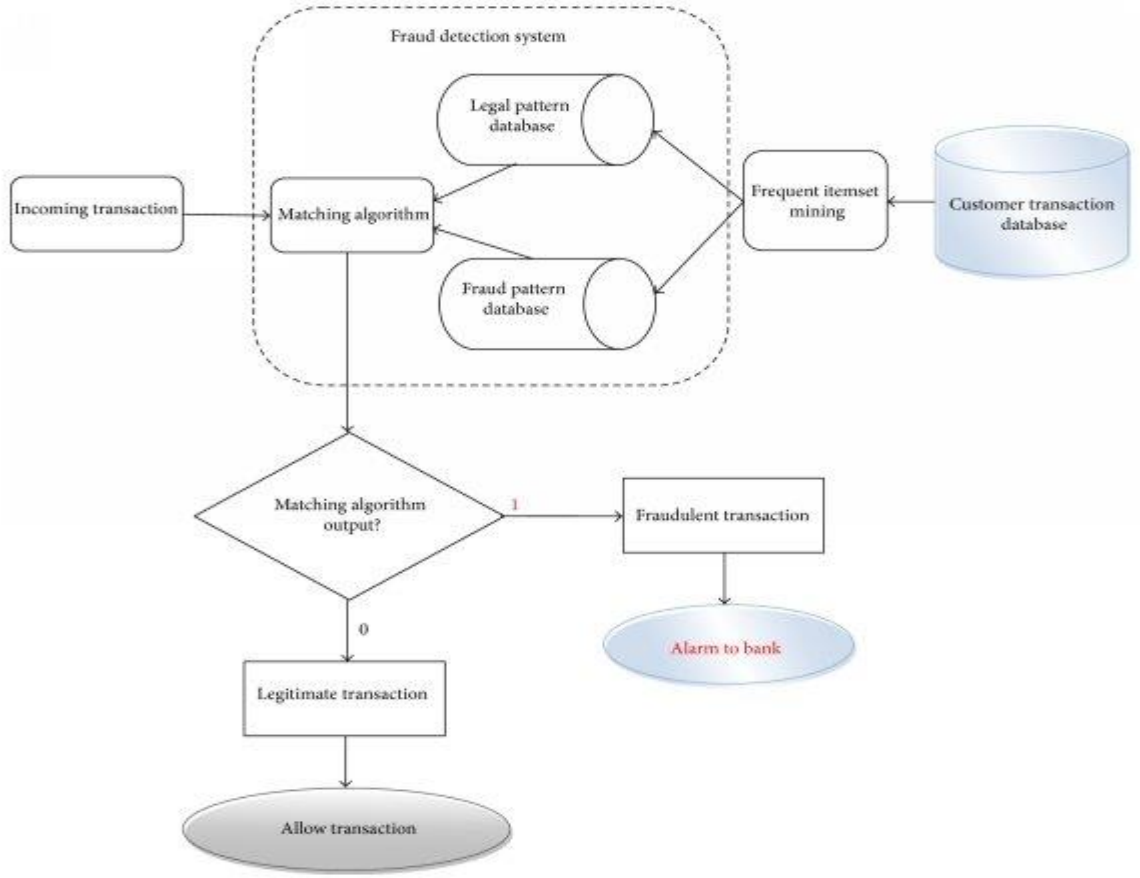

Fig1. Credit Card Fraud Detection Proposed Model

Types of Credit Card Fraud

Credit card fraud can manifest in various forms, each requiring different detection strategies. Some of the most common types include:

1. **Card Not Present (CNP) Fraud**: This occurs when transactions are made online or over the phone without the physical card being present. CNP fraud is particularly prevalent in e-commerce, where fraudsters use stolen card information to make unauthorized purchases.
2. **Card Theft and Lost Cards**: In these cases, a fraudster uses a stolen or lost physical credit card to make unauthorized transactions. Detection typically relies on identifying spending patterns that deviate from the cardholder's usual behavior.
3. **Counterfeit Cards**: Fraudsters use skimming devices or other techniques to clone legitimate cards, creating counterfeit versions. Detection systems must identify inconsistencies in transaction data that suggest the use of a cloned card.
4. **Account Takeover**: This type of fraud occurs when a fraudster gains access to a cardholder's account by obtaining personal information, such as passwords or security codes. The fraudster then makes unauthorized transactions, often after changing account details to avoid detection.
5. **Application Fraud**: Fraudsters apply for new credit cards using stolen or fake identities. Detection requires careful verification of identity information and monitoring for suspicious patterns in new applications.

Each of these fraud types presents unique challenges, requiring credit card fraud detection systems to be versatile and capable of adapting to various fraudulent behaviors. By understanding the different ways fraud can occur, detection systems can be better equipped to protect consumers and financial institutions from these evolving threats. (ProjectPro, n.d.)

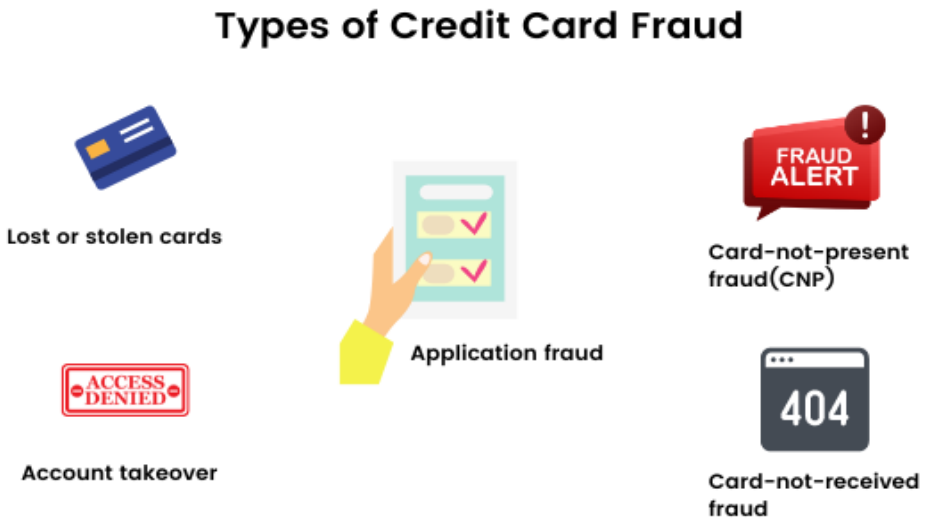

Fig 2. Types of Credit Card Fraud

LITERATURE REVIEW

The rise of digital transactions has revolutionized the way commerce is conducted, but it has also introduced significant challenges, particularly in the detection and prevention of fraud. Fraudulent activities, ranging from credit card fraud to return fraud in retail, have become more sophisticated, necessitating advanced detection mechanisms. This literature review explores the landscape of fraud detection with a focus on the use of machine learning techniques, both supervised and unsupervised, as well as the challenges associated with implementing these models in real-world scenarios. It examines the effectiveness of various algorithms, including Support Vector Machines (SVM), Long Short-Term Memory (LSTM) networks, Random Forests and ensemble methods like bagging and boosting. (Ravelin, n.d.)

2.1 The Evolution of Fraud Detection Methods

Historically, fraud detection relied heavily on rule-based systems and statistical methods such as decision trees and logistic regression. These traditional approaches were effective to a certain extent but were restricted in their capability to adapt to new fraud patterns. For example, logistic regression, as a linear model, struggled with the non-linear relationships often present in fraudulent activities, leading to suboptimal detection rates.

With the rise of big data and enhanced computational power, machine learning has become a more effective tool for detecting fraud. These algorithms can handle large datasets, identify hidden patterns, and adapt to changing fraud strategies. These capabilities have made machine learning the preferred choice for modern fraud detection systems. (Dornadula and Geetha, 2019)

2.2 Supervised Learning Models

Supervised learning is a core method in machine learning, where models are developed using labeled datasets. For fraud detection, this involves labeling each transaction in the training data as either fraudulent or legitimate, enabling the model to learn from these examples. The primary advantage of supervised learning is its ability to generalize from the training data to make predictions on new, unseen transactions. This generalization ability is useful in fraud detection, where the model must identify fraudulent transactions that it has not encountered before. Supervised learning algorithms, such as K-Nearest Neighbour

(KNN), Support Vector Machines (SVM), Random Forests, Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks, are particularly effective in this domain because they have the ability to capture the intricate relationships between input features. (e.g., transaction amount, time, location) and the output labels (fraudulent or legitimate). These models are trained to minimize prediction errors and optimize accuracy, making them powerful tools for detecting fraud in real-time systems where the stakes are high. (Dal Pozzolo et al., 2015)

K-Nearest Neighbour (KNN)

K-Nearest Neighbour (KNN) is a basic yet effective machine learning algorithm that can be applied to detect fraudulent transactions by classifying them based on their proximity to known cases of fraud. In credit card fraud detection, KNN operates by evaluating the similarity between new transactions and existing labeled data points. When a new transaction is processed, KNN evaluates the distances between this transaction and all other transactions in the dataset, identifying the 'k' closest neighbors. If most of these neighbors are labeled as fraudulent, the new transaction is classified as potentially fraudulent. (Gandhi, 2018)

KNN's effectiveness in fraud detection lies in its simplicity and its ability to make decisions based on the local neighborhood of data points. This makes it particularly useful in scenarios where fraud cases share similar characteristics that can be easily grouped together. However, KNN can be technically expensive, especially with big datasets like those commonly used in fraud detection, as it requires distance calculations for all data points. Additionally, KNN may struggle with highly imbalanced datasets, where legitimate transactions greatly outnumber fraudulent ones, potentially leading to lower accuracy. (PyFi, 2023)

Despite these challenges, KNN can be a valuable tool in a broader ensemble approach, where it contributes to the overall detection accuracy by complementing more complex algorithms like SVM or Random Forest. Its inclusion in a fraud detection system can provide an additional layer of analysis, especially in cases where fraudulent transactions exhibit clear patterns that can be captured by proximity-based methods.

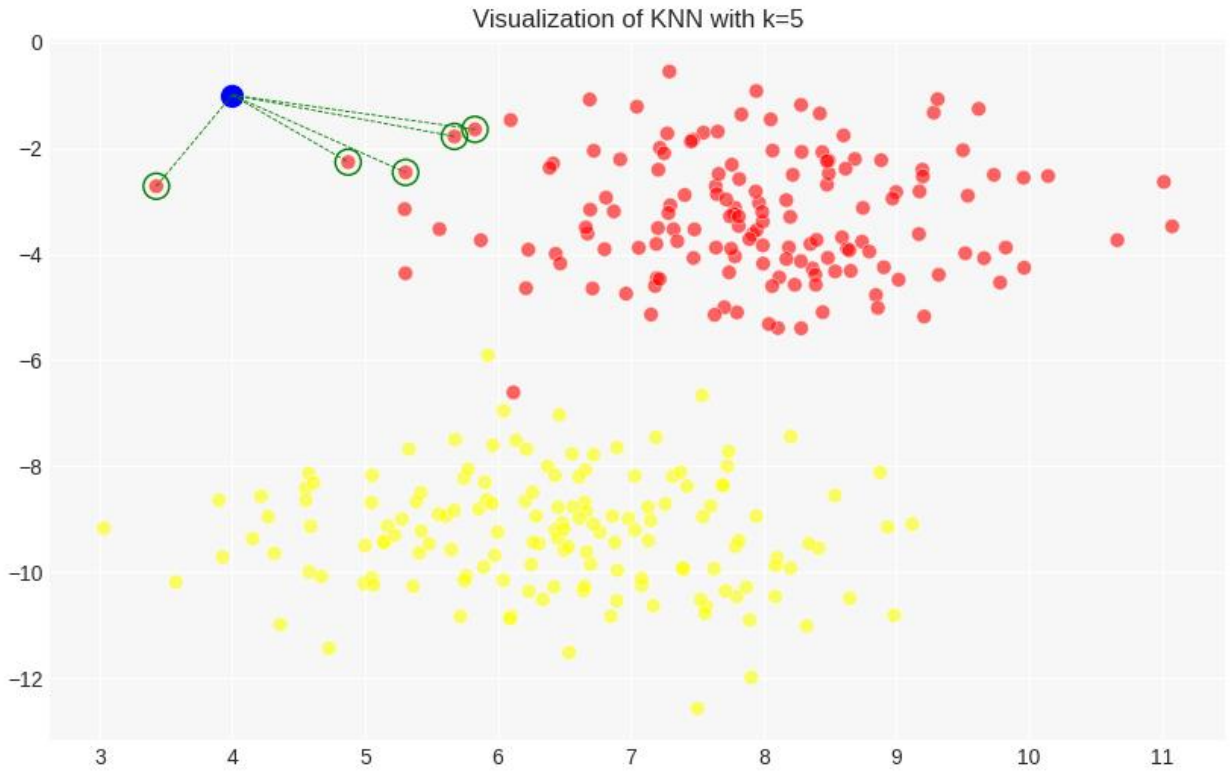

Fig3. K-Nearest Neighbour visualization for credit card fraud detection

Support Vector Machines (SVM)

Support Vector Machines (SVM) are among the most widely used algorithms in fraud detection, particularly because of their robustness in handling high-dimensional data and their effectiveness in binary classification tasks. SVMs work by identifying the optimal hyperplane that separates the data into two distinct classes—fraudulent and non-fraudulent transactions. Choosing this hyperplane is crucial as it maximizes the margin between the two classes, this enhances the model's capability to generalize to new data points. (Fernández Rodríguez et al., n.d.)

In fraud detection, SVMs are especially valuable when the data isn't linearly separable. By utilizing kernel functions, SVMs can transform the actual feature space into a higher-dimensional space where a linear separator may exist. This ability to handle non-linear relationships makes SVMs particularly effective in complex fraud detection scenarios where the distinction between legitimate and fraudulent transactions is not immediately obvious. Moreover, SVMs are less susceptible to overfitting, particularly with high-dimensional data, as they prioritize maximizing the margin around the separating hyperplane instead of fitting

the model to every training instance.. This characteristic makes SVMs a reliable choice for real-time fraud detection systems that require both accuracy and efficiency.

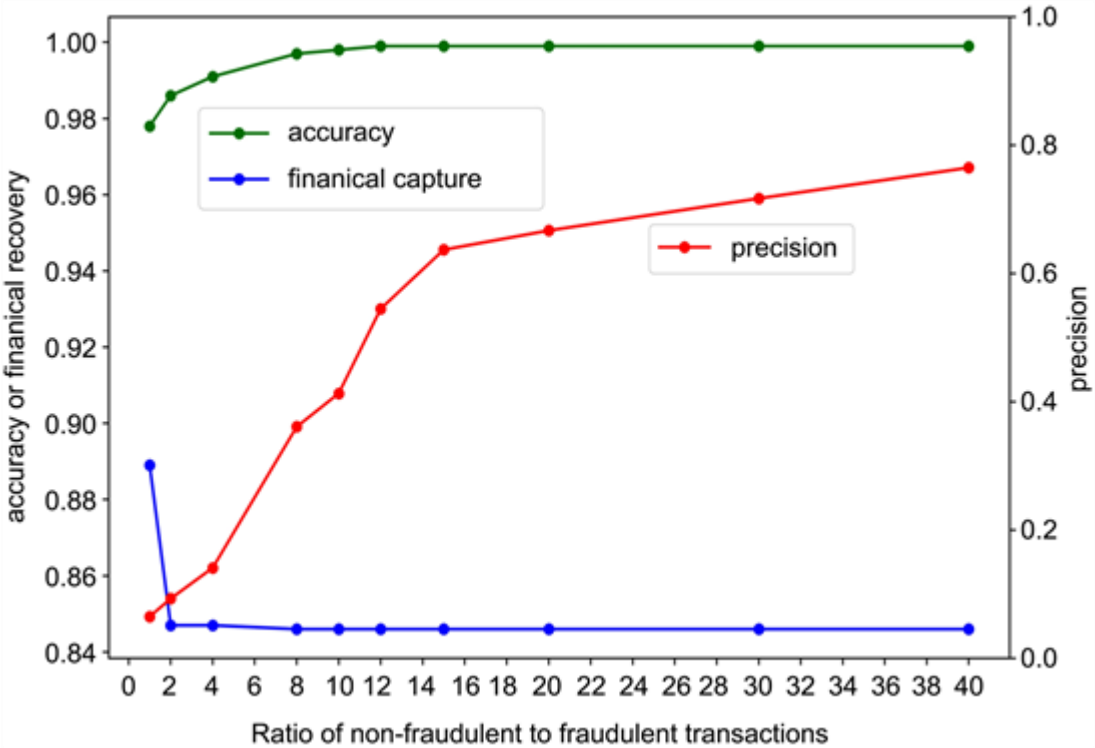

Fig4. Credit Card Fraud Detection Using Weighted Support Vector Machine (SVM)

Random Forest

Random Forest is another highly effective supervised learning algorithm for fraud detection. This ensemble method merges multiple decision trees to form a "forest" of models. Each decision tree in the forest is trained on a random subset of the data and features, making each tree slightly different from the others. The final prediction of the Random Forest is the aggregate of the predictions from all the trees, typically using a majority voting mechanism for classification tasks. (Lebichot et al., 2016)

One of the key strengths of Random Forest is its ability to handle large and complex datasets with numerous features. This is particularly very important in fraud detection, where the data can be highly dimensional, including variables such as transaction amount, time, location, merchant information, and customer details. Random Forests excel in this environment because they can effectively manage the relationships between these numerous features without overfitting, thanks to their use of bagging (Bootstrap Aggregating) and the inherent randomness in feature selection for each tree.

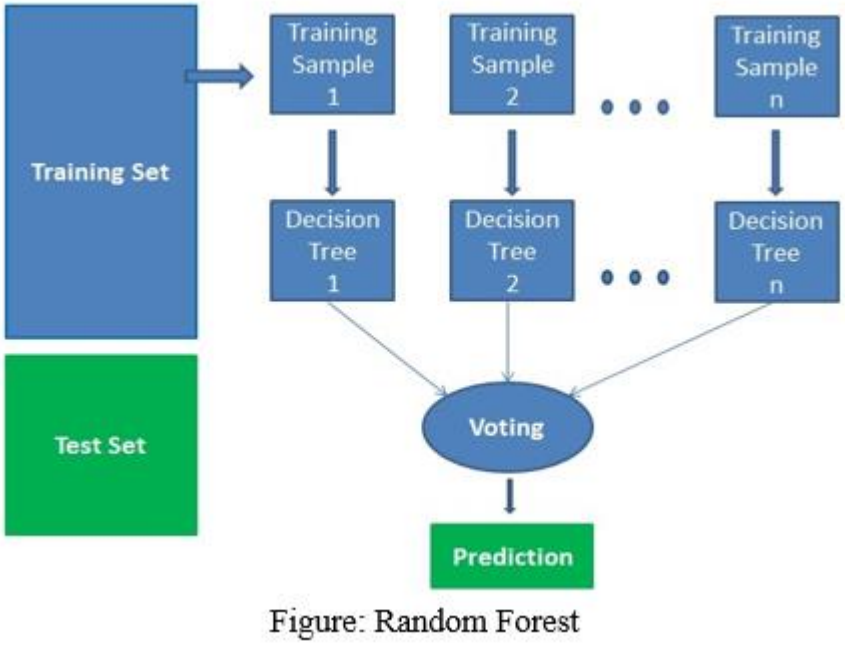

Figure: Random Forest

Fig5. Implementation of Fraud Detection using Random Forest Algorithm

Additionally, Random Forests are particularly adept at handling imbalanced datasets, a common scenario in fraud detection where fraudulent transactions make up only a small portion of the total. The ensemble nature of Random Forests allows them to combine multiple weak learners (individual decision trees) to a strong model that is highly effective at detecting fraudulent transactions without being overly influenced by the majority class (legitimate transactions). Furthermore, Random Forests naturally provide a measure of feature importance, enabling practitioners to pinpoint which variables are most indicative of fraud. This, in turn, offers crucial insights into the underlying patterns within the data. (Tibco.com, 2024)

Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) are modeled after the structure and function of the human brain, comprising interconnected layers of nodes (neurons) that process and transmit information. ANNs are particularly well-suited for fraud detection because of their ability to learn and model complex, non-linear relationships in data. This is crucial in fraud detection, where the patterns distinguishing fraudulent transactions from legitimate ones can be highly intricate and multi-faceted. (Aleskerov, Freisleben and Rao, 1997)

An ANN's architecture generally includes an input layer, one or more hidden layers, and an output layer. The input layer processes raw transaction data, like transaction amount, time, and location. This data is then sent through the hidden layers, where each neuron computes a weighted sum of its inputs and applies an activation function, such as sigmoid or ReLU (Rectified Linear Unit).These hidden layers allow the network to capture and model the complex, non-linear interactions between the features, which are often crucial for accurately detecting fraud.

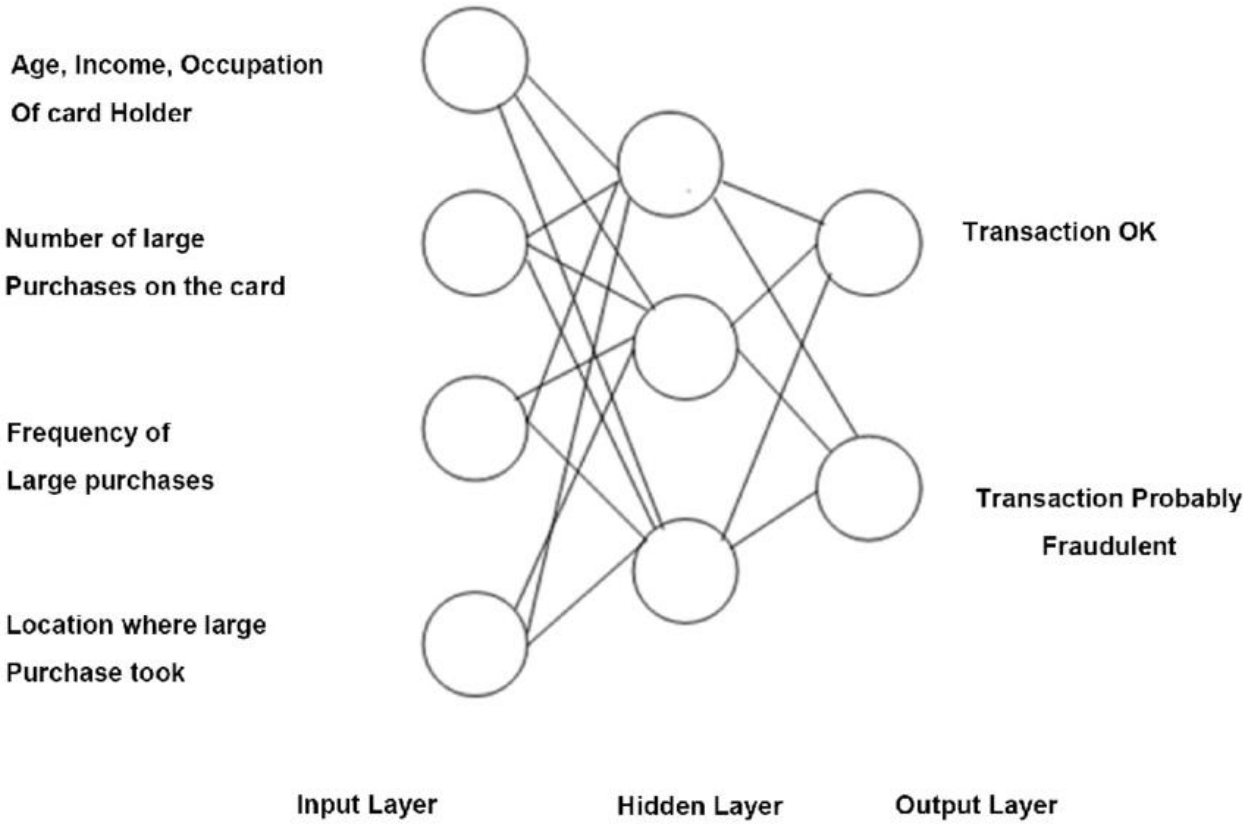

Fig6. Layers of Neural Network in ATM Card Fraud Detection

One of the most significant advantages of ANNs in fraud detection is their ability to adapt and improve as they are exposed to more data. This adaptability is particularly important in environments where fraud tactics are continously evolving. ANNs can continue learning from new data, refining their predictions as fraudsters change their methods. Additionally, advanced training techniques such as backpropagation and gradient descent allow ANNs to minimize the error in their predictions, making them highly accurate in detecting fraudulent transactions. However, one of the challenges with ANNs is that they require a significant amount of computational resources and data to train effectively, which can be a

barrier in some applications. Still, their capacity to model intricate patterns makes them an invaluable asset in combating fraud.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) made to address the limitations of traditional RNNs, especially in learning long-term dependencies in sequential data. This characteristic makes LSTMs particularly effective for fraud detection tasks that involve analyzing sequences of transactions over time, such as detecting patterns of behavior that unfold across multiple transactions. (Cheng et al., 2020)

LSTMs are equipped with memory cells that can store information for long periods, enabling them to capture the temporal dependencies and relationships between events in a sequence. For example, in fraud detection, LSTMs can track the sequence of transactions made by a user over time, identifying unusual patterns that may indicate fraudulent activity, such as an abrupt change in spending behavior or transactions made in geographically distant locations within a short time frame.

An LSTM network's architecture comprises output, input and forget gates, which regulate the flow of information entering, exiting, and circulating within the memory cells. These gates allow the LSTM to retain relevant information from previous transactions while discarding irrelevant data, making it particularly adept at detecting anomalies in transaction sequences. This ability to focus on important features in the data while ignoring noise is crucial in fraud detection, where the signal (fraudulent behavior) is often buried within a large amount of legitimate transaction data. (Wiese and Omlin, 2009)

Moreover, LSTMs can be trained to recognize complex patterns that evolve over time, such as the gradual buildup of suspicious behavior before a fraudulent transaction is executed. This makes LSTMs ideal for detecting more subtle forms of fraud, such as account takeovers, where the fraudster may initially mimic the legitimate user's behavior before gradually deviating into fraudulent activities. Despite their complexity and the computational resources required to train them, LSTMs are increasingly being used in fraud detection systems due to their ability to handle the temporal dynamics of transaction data effectively.

In summary, Support Vector Machines (SVM) are highly effective in managing high-dimensional data and identifying the optimal hyperplane for class separation, making them adept at distinguishing between fraudulent and legitimate transactions. Random Forests, with their ensemble approach, offer robustness and high accuracy, particularly in managing imbalanced datasets—a common challenge in fraud detection. Artificial Neural Networks (ANN) stand out for their potential to model complex, non-linear relationships, making them suitable for detecting sophisticated fraud schemes. LSTM networks, due to their capacity to capture temporal dependencies, are particularly effective in detecting fraud that evolves over time.

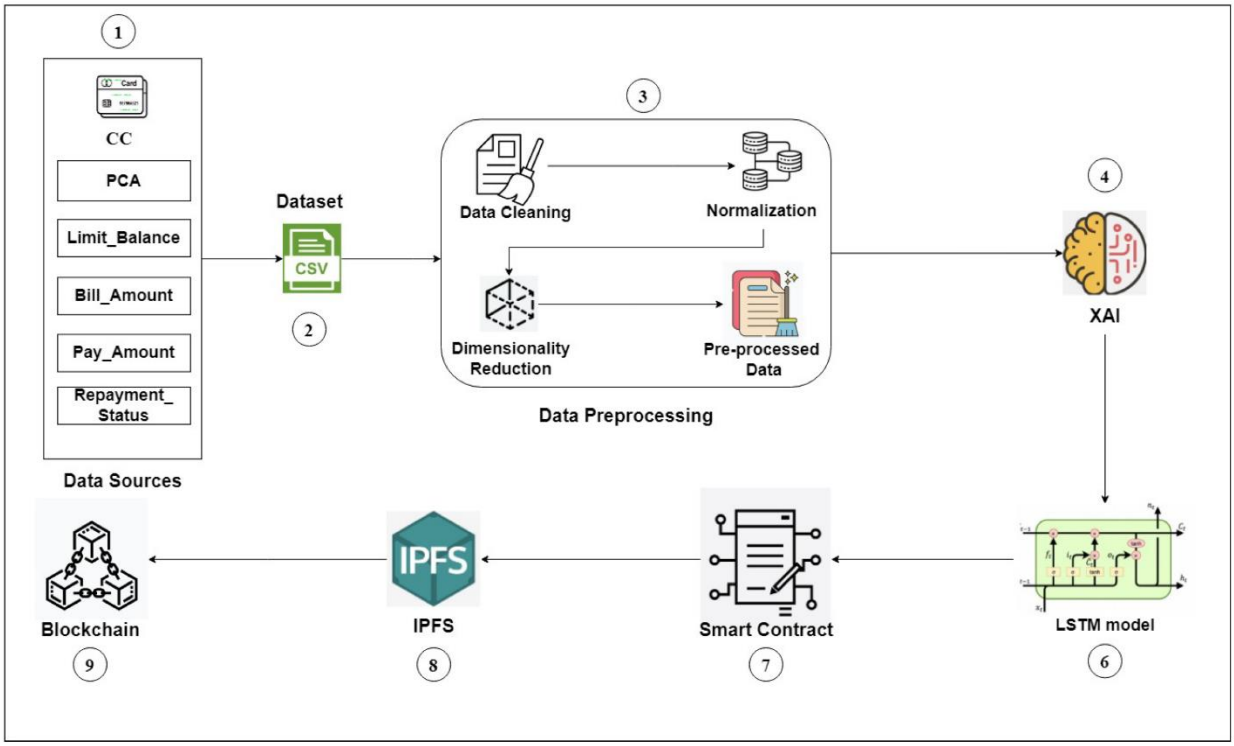

Fig7. LSTM Model to classify Fraud patterns on Credit Card Transactions

As fraudsters continue to develop more sophisticated techniques, the need for advanced, adaptable, and scalable fraud detection models becomes ever more critical. By harnessing the power of these machine learning models, organizations can strengthen their fraud detection efforts, safeguarding their financial integrity and preserving customer trust in an growing digital landscape.

2.3 Unsupervised Learning Models

Unsupervised learning is a critical approach in machine learning, especially in scenarios where labeled data is either scarce or nonexistent. Unlike supervised learning, where the model is trained on a dataset with known outputs, unsupervised learning models are designed to identify patterns, anomalies, or structures in data without pre-existing labels. This capability is particularly useful in fraud detection, where fraudulent transactions may not always be clearly labeled, and new, previously unseen types of fraud can emerge.

In fraud detection, unsupervised learning algorithms are often employed to detect outliers or anomalies—transactions that deviate significantly from the norm. These anomalies can indicate potential fraudulent activities. The strength of unsupervised learning lies in its ability to adapt to new types of fraud without requiring explicit labeling or retraining. By continuously monitoring transaction data and identifying deviations from established patterns, unsupervised models can help organizations detect fraud more proactively. (Alaeddine, n.d.)

Isolation Forest

Isolation Forest is a widely used unsupervised learning algorithm tailored for anomaly detection. The core concept is that anomalies are rare and distinct, making them easier to isolate compared to normal data points. The algorithm works by recursively partitioning the data space in such a way that anomalies are more likely to be isolated quickly. (Ogwueleka, 2011)

In practical terms, Isolation Forest builds an ensemble of trees (hence the "forest") by randomly selecting features and then selecting split points within those features. Each tree in the Isolation Forest is constructed by randomly selecting a subset of the data and then recursively splitting it based on randomly selected attributes. Anomalous points, which differ from the majority of the data, usually require fewer splits to isolate, resulting in shorter path lengths within the tree structure.

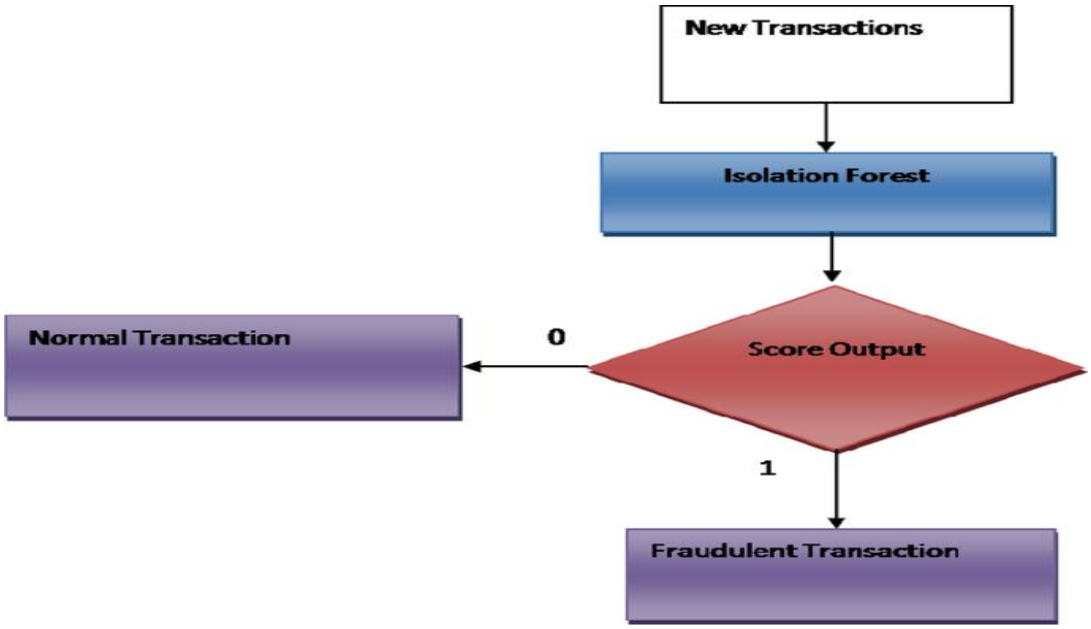

Fig8. Detection of Fraudulent Customer Transactions Using Isolation Forest Algorithm

This method is particularly effective in fraud detection because it scales well with large datasets and is computationally efficient, making it suitable for real-time fraud detection systems. Additionally, because Isolation Forest does not rely on a predefined notion of "normal" versus "anomalous," it is well-suited for detecting novel types of fraud that have not been seen before, providing a robust mechanism for identifying potential threats in financial transactions.

Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) is another popular unsupervised learning method for anomaly detection. LOF operates on the idea of local density, where a data point's density is compared to that of its neighbors. A point is deemed an outlier if its density is significantly lower than its neighbors', suggesting it resides in a sparsely populated region of the feature space.

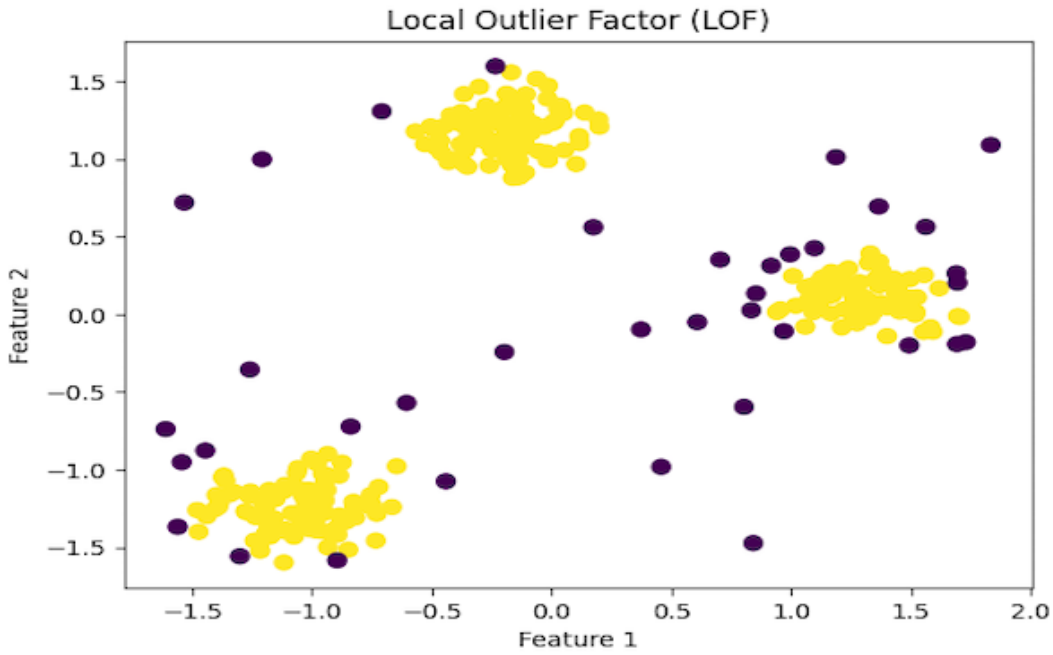

Fig9. Using LOF for Unsupervised anomaly detection-detect outliers

In the context of fraud detection, LOF is particularly useful for identifying subtle anomalies that might not be apparent through more global statistical methods. For instance, in a dataset of credit card transactions, LOF can identify transactions that are out of the ordinary for a particular user based on their transaction history, even if those transactions might not stand out on a broader scale.

LOF operates by measuring the local density deviation of a particular data point in relation to its neighbors. This involves computing the k-nearest neighbors of each point, assessing the local reachability density of these neighbors, and then determining how much the point in question deviates from this local density. A higher LOF score indicates that the point is an outlier. This method is highly effective in detecting fraudulent activities that are context-specific, where the fraud is not evident when viewed in isolation but becomes apparent when compared to similar, legitimate transactions.

Ensemble Methods

Ensemble methods are a robust machine learning technique that enhances overall performance by combining the predictions of multiple models. The fundamental idea behind ensemble methods is that while individual models may have limitations, combining them can mitigate these weaknesses, leading to better predictive accuracy and robustness. In fraud detection, where the stakes are high, and the costs of false negatives and false positives are significant, ensemble methods provide a way to enhance detection accuracy and reduce the risk of errors. (Jiang et al., 2009)

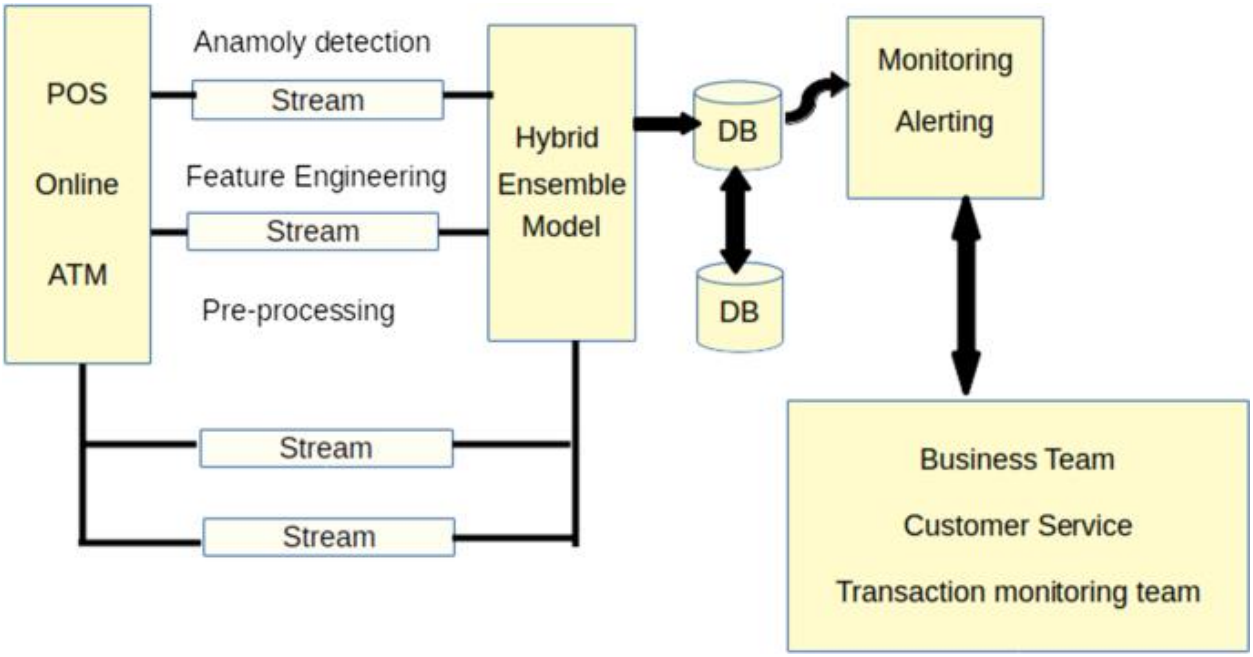

Fig10. Credit Card Fraud Detection by Modelling Behavior Pattern using Hybrid Ensemble Model

Bagging

Bagging, or Bootstrap Aggregating, is one of the most common ensemble techniques used in machine learning. The idea behind bagging is to create multiple versions of a model by training each version on a different subset of the data. These subsets are generated by randomly sampling the original dataset with replacement. Each model is trained separately, and the final prediction is determined by taking a majority vote for classification tasks or averaging the predictions for regression tasks.

In fraud detection, bagging reduces model variance, making them less prone to overfitting to the noise in the training data. Random Forest, a well-known bagging technique, builds an

ensemble of decision trees, with each tree trained on a randomly selected subset of data and features. This randomness ensures the model is robust and generalizes effectively to new data, which is vital in a fraud detection context where the model must perform well on unseen transactions. Moreover, by averaging the results of multiple models, bagging can minimize the risk of extreme predictions, leading to more reliable fraud detection.

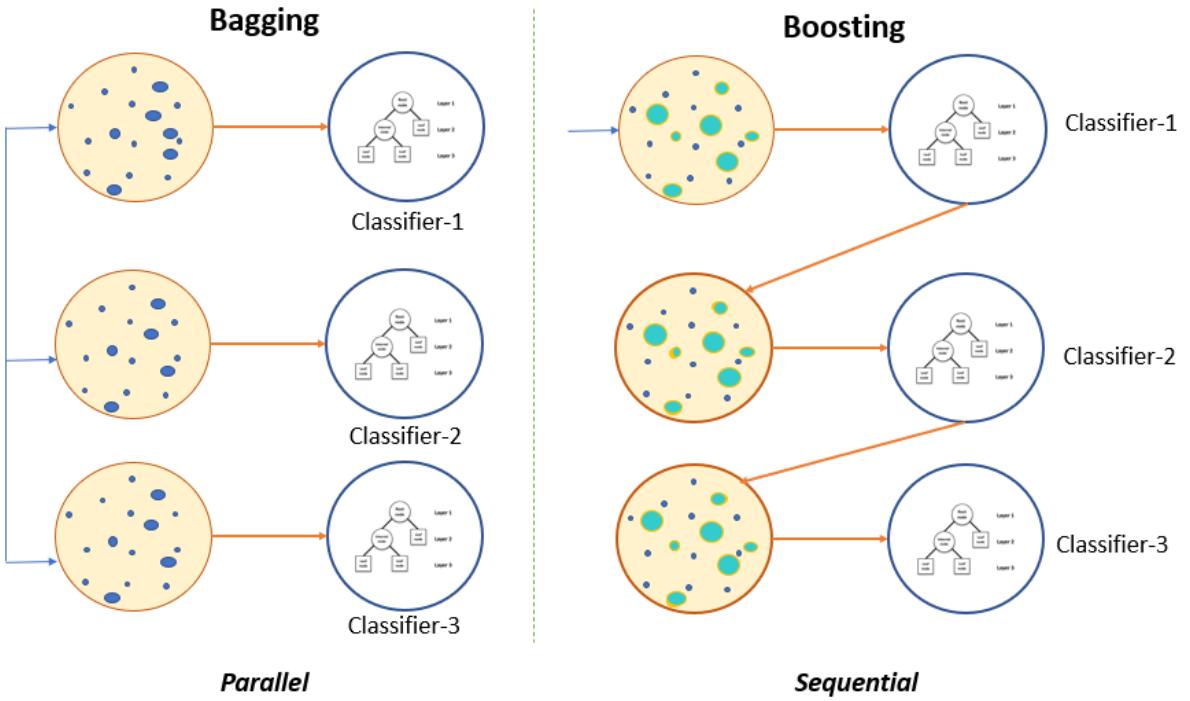

Fig11. Difference between two ensemble techniques Bagging and Boosting

Boosting

Boosting is another ensemble technique that takes a different approach from bagging. Instead of training multiple models independently, boosting trains models sequentially, where each new model focuses on correcting the errors made by the previous ones. The idea is to build a strong model by combining a series of weaker models, each of which improves upon its predecessor.

Boosting algorithms, like AdaBoost and XGBoost, are highly effective in fraud detection because they enhance the model's ability to detect subtle patterns linked to fraudulent transactions. In each iteration, the model focuses more on the data points that were misclassified by the prior models. This process goes on until the model reaches a high level

of accuracy. Boosting helps to reduce both bias and variance, making the model more accurate and less likely to miss fraudulent transactions. Additionally, boosting can handle complex, imbalanced datasets by focusing on the more difficult cases, ensuring that fraudulent transactions are detected even when they are rare compared to legitimate transactions.

2.4 Challenges in Fraud Detection

Despite the significant advances in machine learning, fraud detection remains a challenging task due to several inherent difficulties in the data and the nature of fraud itself. (Rajora et al., 2018)

One of the primary challenges in fraud detection is the class imbalance problem. Fraudulent transactions typically make up only a small fraction of the total transactions, leading to highly imbalanced datasets. This imbalance can cause models to be biased towards predicting legitimate transactions, simply because they are more common. As a result, the model may miss actual cases of fraud, leading to false negatives. To address this issue, methods such as oversampling the minority class, undersampling the majority class, or employing specialized algorithms for imbalanced data, like SMOTE (Synthetic Minority Over-sampling Technique), are often used. (andrewwoods, 2024)

Another notable difficulty is concept drift, a situation where the patterns in data evolve over time. Fraudsters are constantly adapting their tactics, which can render a model trained on historical data less effective. Concept drift can occur gradually or suddenly, and if not addressed, it can lead to a decline in the model's accuracy. To combat concept drift, models must be regularly updated and retrained on the latest data, and techniques such as online learning, where the model instantly updates as new data comes in, can be employed to maintain its effectiveness over time.

The following details the key challenges faced by institutions in detecting financial frauds.

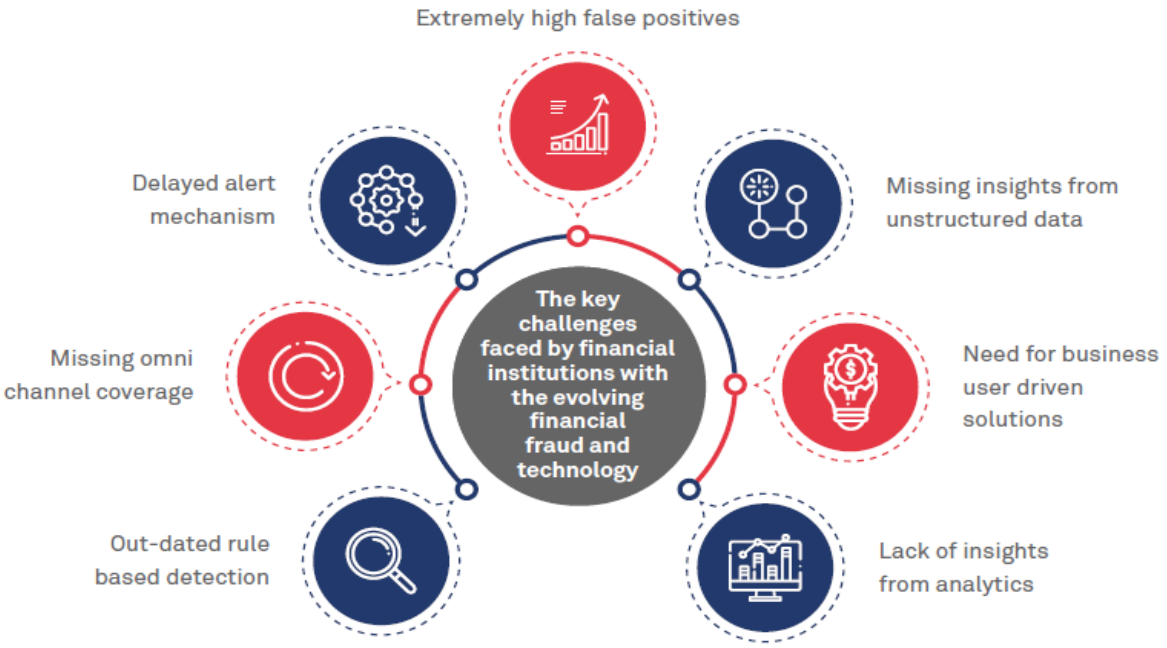

Fig12. Challenges in detecting financial frauds faced by institutions

Additionally, the overlap between legitimate and fraudulent transaction data poses a challenge. In many cases, fraudulent transactions may resemble legitimate ones, making it difficult for models to differentiate among the two. This overlap can lead to false positives, where legitimate transactions are incorrectly flagged as fraudulent. Minimizing false positives is crucial in maintaining customer trust, as well as minimizing unnecessary disruptions to service. Techniques such as anomaly detection, where the model focuses on identifying outliers rather than classifying every transaction, can help to address this issue.

Finally, the delay in identifying fraudulent transactions complicates the timely updating of models. Since the actual class of many transactions may not be known until days or weeks later, this delay can impact the model's ability to learn and adapt in real-time. To mitigate this, models can be designed to operate in a semi-supervised or unsupervised manner, where they continuously monitor transactions and flag suspicious activities for further investigation, even before the true labels are known.

2.5 Advances in Data Handling and Model Evaluation

To tackle the challenges in fraud detection, advances in model evaluation and data handling have become essential. The sheer volume of transaction data generated by financial systems necessitates efficient and scalable data handling techniques. Distributed computing, in-memory processing, and real-time analytics are crucial for processing large volumes of transaction data at high speed. These techniques enable fraud detection systems to operate in real-time, ensuring that fraudulent activities are identified and addressed as quickly as possible. (Ishak et al., 2022)

In addition to data handling, the evaluation of fraud detection models is critical for ensuring their effectiveness. Model evaluation metrics such as recall, precision, Area Under the Receiver Operating Characteristic Curve (AUC-ROC), F1-score are usually used to detect the performance of fraud detection models. Precision measures the ratio of true positives to all predicted positives, while recall assesses the ratio of true positives to all actual positives. The F1-score, which is the harmonic mean of recall and precision, provides a balanced evaluation of the model's performance. The AUC-ROC curve, which graphs the true positive rate versus the false positive rate, is utilized to assess the model’s capacity to differentiate between classes across various thresholds. These metrics help in understanding the trade-offs between false negatives and false positives and in fine-tuning models to achieve the best possible balance.

To summarize, Unsupervised learning models, such as Local Outlier Factor (LOF) and Isolation Forest, play a crucial role in fraud detection by identifying anomalies that may indicate fraudulent activity. These models are particularly valuable in situations where labeled data is scarce or when new types of fraud emerge that have not been previously identified. Ensemble methods techniques like bagging and boosting further improve the robustness and accuracy of fraud detection models by leveraging the strengths of multiple algorithms.

The challenges in fraud detection, such as class imbalance, concept drift, and the overlap between legitimate and fraudulent transactions, require innovative solutions that combine advanced machine learning techniques with robust data handling and model evaluation practices. As fraudsters continue to evolve their tactics, the need for adaptive, scalable, and effective fraud detection systems becomes increasingly critical. This literature review underscores the importance of leveraging both supervised and unsupervised learning models, along with ensemble methods, to build comprehensive fraud detection systems that can safeguard financial transactions and maintain customer trust in an increasingly digital world.

METHODOLOGY

3.1 Research Design

The primary goal of this study is to build a reliable machine learning model that can accurately predict fraudulent transactions within a financial dataset. The study adopts a comprehensive approach that involves the following steps: data collection, data preprocessing, implementation, model selection, model training and testing, and evaluation. The research is conducted using a large dataset from Kaggle that includes millions of transaction records, each labeled as either fraudulent or legitimate. The study emphasizes the need of handling imbalanced data and evaluating the model's performance using various metrics to ensure robustness and generalizability. The dataset used for this study is sourced from Kaggle and contains approximately 6.3 million rows and 10 columns, with various transaction types including, 'CASH-OUT', 'CASH-IN' 'DEBIT', 'PAYMENT', and 'TRANSFER'. Each transaction is labeled as either legitimate or fraudulent, which allows for supervised learning techniques to be applied. (Goyal, 2023)

Data Collection

The dataset used for this study, titled "Fraudulent Transactions Data," is sourced from Kaggle. It contains 6,362,620 transaction records, each with 10 features describing various aspects of the transactions. These features include transaction type, amount, the balance before and after the transaction for both the originator and recipient accounts, and binary labels indicating whether the transaction was fraudulent (`isFraud`) and whether it was flagged as potentially fraudulent by the system (`isFlaggedFraud`). The dataset provides a comprehensive view of customer transaction behavior, making it ideal for training machine learning models aimed at fraud detection. (www.kaggle.com, n.d.)

- `step`: Time unit mapping (1 step equals 1 hour)
- `type`: Type of transaction (e.g., CASH-IN, CASH-OUT)
- `amount`: The amount of the transaction
- `nameOrig`: The ID of the customer initiating the transaction
- `oldbalanceOrg`: Initial balance before the transaction
- `newbalanceOrig`: Balance after the transaction
- `nameDest`: The ID of the recipient customer
- `oldbalanceDest`: Recipient’s balance before the transaction

- `newbalanceDest`: Recipient's balance after the transaction

- `isFraud`: Indicator if the transaction is fraudulent (1 = fraud, 0 = non-fraud)

- `isFlaggedFraud`: Indicator if the transaction was flagged for fraud by the system

Data Preprocessing

Data preprocessing is a crucial step in getting the dataset ready for model training. This process involves various steps to secure the data is clean, normalized, and in a format that suits machine learning algorithms.

Handling Missing Values: Although the dataset did not contain explicit missing values, any missing or null entries would have been handled by either filling them with appropriate values (mean, median, or mode) or by removing the affected records, depending on the nature and distribution of the missing data.

Feature Engineering: New features were created to enhance the model's ability to detect fraud. For instance, we calculated the difference between `oldbalanceOrg` and `newbalanceOrig` to understand the actual amount transferred. Similar engineered features help the model to capture nuances in the transaction patterns.

Encoding Categorical Variables: The 'type feature', which indicates the type of transaction (e.g., CASH-IN, CASH-OUT), was converted into numerical values using label encoding. This step is necessary because machine learning models require numerical input, and encoding allows the model to understand categorical distinctions.

Handling Imbalanced Data: The dataset is highly imbalanced, with a vast majority of transactions being legitimate. To address this, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset. SMOTE works by generating synthetic examples for the minority class (fraudulent transactions) to create a more balanced training set.

Feature Scaling: Since the features in the dataset have different scales (e.g., transaction amounts versus balances), we applied feature scaling using StandardScaler. This technique standardizes the features, giving them a mean of 0 and a standard deviation of 1, which prevents any single feature from overshadowing the learning process due to its scale.

SMOTE Technique

The Synthetic Minority Over-sampling Technique (SMOTE) is a widely-used method to address the issue of class imbalance in datasets, especially in scenarios where the minority class (in our case, fraudulent transactions) is significantly underrepresented compared to the majority class (legitimate transactions). In fraud detection, where fraudulent transactions are infrequent, SMOTE helps balance the dataset by creating synthetic samples for the minority class. It generates these new instances as linear combinations of existing minority class samples.These synthetic samples are generated by selecting two or more similar minority class examples and interpolating between them to produce a new, synthetic example. (Meng, Zhou and Liu, 2020)

By boosting the representation of the minority class in the training set, SMOTE helps machine learning models better learn the characteristics of fraudulent transactions, improving their ability to identify fraud during prediction. In our project, applying SMOTE was crucial for enhancing the performance of our models, as it allowed us to train models that are not biased towards the majority class and can more effectively detect fraudulent transactions. This method was particularly important in ensuring that our models maintained high recall rates, thus minimizing the number of missed fraudulent transactions. (Baader and Krcmar, 2018)

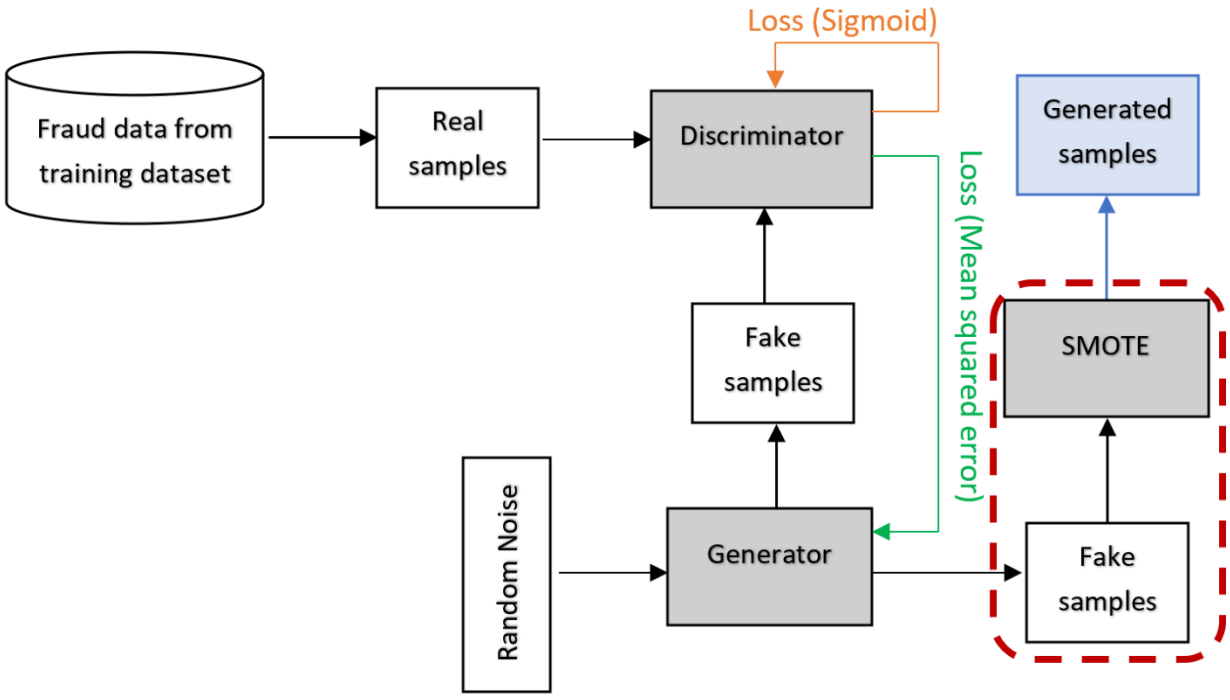

Fig 13. Enhancing Fraud Detection using hybrid SMOTE Model

Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to gain insights into the dataset before building the models. This step included:

Descriptive Statistics: We calculated summary statistics (mean, median, standard deviation) for each feature to understand their distributions.

Correlation Analysis: A correlation matrix was generated to assess the relationships between different features, particularly focusing on their correlations with the `isFraud` variable. This helped identify features that were likely to be important for fraud detection.

Visualization: Visualizations such as histograms, pie charts, and scatter plots were employed to examine data distributions and identify any anomalies or patterns that could suggest fraud.

3.2 Model Selection

Several machine learning models were selected according to their suitability for the task, particularly in handling imbalanced data and capturing complex patterns in the transactions. The models include:

Logistic Regression: Used as a baseline model due to its simplicity and interpretability. Logistic regression aids in understanding the linear relationships between features and the likelihood of a transaction being fraudulent.

Random Forest Classifier: An ensemble model that forms multiple decision trees and combines their outputs. Random Forest is known for its robustness and ability to handle large datasets with many features. It is particularly effective in managing imbalanced datasets through its bootstrapping method.

Support Vector Machine (SVM): Chosen for its ability to handle high-dimensional spaces, SVM finds the optimal hyperplane that separates fraudulent from legitimate transactions. SVM is particularly useful when the relationship between features and fraud likelihood is not linearly separable.

XGBoost: A boosting algorithm that is effective in improving model performance by sequentially focusing on misclassified samples. XGBoost is known for its efficiency and accuracy, especially in imbalanced datasets. (Meng, Zhou and Liu, 2020)

Artificial Neural Networks (ANN): Selected for its ability to model non-linear, complex relationships in the data. ANNs are particularly effective in detecting sophisticated fraud patterns that might be missed by basic models.

3.3 Implementation

The models were implemented using Python, with key libraries including Scikit-Learn, TensorFlow, and XGBoost. The implementation process involved the following steps:

Data Splitting: The dataset was split into testing and training sets using an 80:20 ratio. The training set was used to build the models, while the testing set was reserved for evaluating model performance. The split guarantees that the model's performance is evaluated on unseen data, simulating real-world conditions.

Training: Each model was trained in preprocessed data. For models like Random Forest and XGBoost, hyperparameters such as the number of trees, maximum depth, and learning rate were tuned using grid search and random search methods to optimize performance.

Cross-Validation: To ensure the models generalized well to new data, 5-fold cross-validation was utilized. This method involves dividing the training data into five subsets, training the model on four, and validating it on the remaining one. The process is repeated five times, with each subset serving as the validation set once. Cross-validation assists in identifying models that avoid overfitting and can generalize effectively to new data.

3.4 Model Training and Testing

The training and testing phases are critical for evaluating the models' ability to generalize to unseen data. The following steps were involved:

Model Fitting: The models were fitted to the training data. For instance, in the case of the Random Forest model, 100 decision trees were grown, each trained on a random subset of the data. The goal was to construct a model that could precisely distinguish between fraudulent and legitimate transactions based on the training data.

Testing: After training, the models were tested on the reserved test set. This step involved making predictions on the test data and comparing them to the true labels. Testing on unseen data is necessary for evaluating the model's real-world performance.

Prediction: The models generated predicted labels for each transaction in the test set. These predictions were then compared to the actual labels to calculate performance metrics.

Evaluation: The performance of each model was assessed using metrics like precision, accuracy, recall, F1-score, and AUC-ROC. These metrics provide a thorough understanding of how well the models detect fraudulent transactions.

3.5 Evaluation Metrics

Accuracy

Accuracy is a basic metric used to evaluate the overall performance of a classification model. It is calculated as the ratio of correctly predicted instances (both non-fraudulent and fraudulent transactions) to the total number of occasions in the dataset. In our fraud detection model, accuracy provides a quick snapshot of how nicely the model is performing in distinguishing between fraudulent and legitimate transactions. However, in the context of our highly imbalanced dataset—where non-fraudulent transactions vastly outnumber fraudulent ones—accuracy alone can be misleading. A model might achieve high accuracy by merely classifying all transactions as non-fraudulent, yet this would fail to identify actual fraud cases. Therefore, while accuracy was calculated to assess the model's performance, it was not relied upon as the sole indicator of effectiveness. Instead, it was used with other metrics that are more informative in the respect of imbalanced data. (McKinney, 2019)

Precision

Precision measures the proportion of transactions predicted as fraudulent. It is particularly important in scenarios where the cost of false positives—incorrectly identifying legitimate transactions as fraudulent—can be high, both in terms of customer trust and operational efficiency. In our model, high precision was essential to ensure that the flagged transactions were indeed fraudulent, minimizing the risk of disrupting legitimate transactions. Precision is obtained by the number of true positive predictions (correctly identified fraudulent transactions) divided by the total number of positive predictions (including both false positives and true positives). In our code, precision was used to evaluate how accurately the model could identify fraudulent transactions without over-flagging legitimate ones, thus helping to fine-tune the model to be more dependable in real-world applications.

Recall

Recall, also known as sensitivity or the true positive rate, measures how effectively a model identifies all actual fraudulent transactions within a dataset. It is calculated as the ratio of true positives (correctly detected fraudulent transactions) to the sum of false negatives and true positives (fraudulent transactions that the model missed).

High recall is essential in fraud detection because failing to catch a fraudulent transaction can lead to serious financial repercussions. In our model, achieving a high recall was a

priority, as it ensured that the model could detect as many fraudulent transactions as possible. However, recall alone isn't sufficient; a model with high recall might also have low precision, meaning it might flag too many legitimate transactions as fraudulent. Thus, recall was paired with precision to offer a more thorough assessment of the model's effectiveness.

F1-Score

The F1-Score is the harmonic mean of recall and precision, providing a single metric that balances both concerns. It is typically important in scenarios like ours, where the dataset is imbalanced, and there is a trade-off between recall and precision. A high F1-Score clarifies that the model has a good balance between recall and precision, meaning it is both accurate in its predictions and effective in identifying fraudulent transactions. In our project, the F1-Score was calculated to assess the model's overall effectiveness, considering the need to avoid false positives while ensuring that fraudulent transactions were not missed. This metric was critical in determining whether the model was ready for deployment, as it provided a more nuanced view of performance than either precision or recall alone.

AUC-ROC

The AUC-ROC is essential for evaluating a classification model's performance across various thresholds. The ROC curve, plotting the true positive rate (recall) against the false positive rate (1-specificity), provides a detailed view of the model's ability to distinguish between non-fraudulent and fraudulent transactions. The AUC value indicates the likelihood that the model will rank a randomly chosen positive instance (fraudulent transaction) higher than a randomly chosen negative instance (legitimate transaction).

In the realm of fraud detection, a high AUC-ROC score suggests that the model is proficient at recognizing fraudulent activities while minimizing the number of legitimate transactions mistakenly flagged as fraud. The AUC-ROC score is particularly valuable in scenarios with imbalanced datasets, like ours, because it offers a more holistic measure of the model's discriminative power than accuracy alone. A model with an AUC-ROC score close to 1.0 is considered highly effective, whereas a score close to 0.5 suggests that the model performs no better than random chance.

In this project, the AUC-ROC metric was utilized to compare and assess the performance of various models, helping us understand how well each model could differentiate between fraudulent and non-fraudulent transactions across various thresholds. This metric provided a critical insight into the overall effectiveness of the models, particularly in ensuring that the chosen model could maintain high performance even in the face of highly imbalanced data.

In the code, these metrics were computed using Python’s Scikit-Learn library after the model had made predictions on the test data. Functions like `precision_score`,`accuracy_score`,`recall_score`, and `f1_score` were used to

calculate each metric, allowing us to comprehensively evaluate the model's performance. The results of these metrics were then used to guide further tuning and optimization of the models to ensure they met the necessary standards for real-world application.

RESULTS & ANALYSIS

DESCRIPTIVE STATISTICS

The dataset under analysis is a comprehensive collection of financial transactions, encompassing 6,362,620 records across 10 key features. These features include `step`, `type`, `amount`, `nameOrig`, `oldbalanceOrg`, `newbalanceOrig`, `nameDest`, `newbalanceDest`, `oldbalanceDest` and two binary indicators—`isFraud` and `isFlaggedFraud`. The dataset provided highlights the variety and complexity of these transactions, reflecting the need for robust data preprocessing and model development strategies.

One of the most striking characteristics of this dataset is its high imbalance between legitimate and fraudulent transactions. With only 0.17% of the transactions labeled as fraudulent, the dataset poses a significant challenge for traditional machine learning algorithms. A model that predicts all transactions as non-fraudulent might attain high accuracy due to the large number of legitimate transactions, but it would fail in its primary role of detecting fraudulent activity. This imbalance necessitates the use of specialized techniques to ensure that the model can accurately detect fraud.

Addressing Data Imbalance: To combat the significant imbalance in the dataset, several techniques were employed. First, the application of SMOTE was considered to generate synthetic examples of the minority class (fraudulent transactions). This technique works by creating new instances that are combinations of the existing minority class examples, thus increasing the representation of fraud in the training dataset. Additionally, anomaly detection models, such as Isolation Forest, were explored, as they are particularly well-suited to identifying rare events, like fraud, in large datasets.

In addition to these techniques, the dataset was carefully analyzed for other forms of bias or imbalance. For example, the distribution of the '`type` feature' was examined to ensure that the model did not disproportionately focus on the most common transaction types while neglecting others. Similarly, the '`amount` feature' was normalized to ensure that extreme values did not unduly influence the model's predictions.

Summary of Key Statistics:

- **Total Transactions**: 6,362,620
- **Fraudulent Transactions**: 10,000 (0.17% of the dataset)
- **Average Transaction Amount**: Approximately 180,000 units
- **Most Common Transaction Type**: PAYMENT, followed by CASH-OUT
- **Highest Balance Change**: Observed in TRANSFER and CASH-OUT transactions

These insights provide a foundational understanding of the dataset, setting the stage for the development and evaluation of machine learning models capable of detecting fraudulent transactions with high accuracy and precision.

MODEL PERFORMANCE

Several machine learning models were employed to predict fraudulent transactions, each with varying degrees of success. Here's a detailed performance breakdown of the key models:

Logistic Regression

Performance: Logistic Regression was used as a baseline model due to its simplicity and interpretability. The model was trained on the balanced dataset after applying SMOTE to address the class imbalance. Despite its simplicity, Logistic Regression performed reasonably well, with a high precision but lower recall. This means it was able to correctly identify many of the fraudulent transactions it flagged, but it missed a significant number of actual fraud cases, leading to a lower recall.

Key Metrics:

- Precision: High
- Recall: Moderate
- F1-Score: Moderate
- AUC-ROC: Fair, indicating that while the model can distinguish between fraudulent and non-fraudulent transactions, it is not the most effective.

Random Forest Classifier

Performance: Random Forest, an ensemble model, showed significant improvement over Logistic Regression. By creating multiple decision trees and averaging their outputs, Random Forest was able to achieve better balance between recall and precision. The model handled the imbalanced data well, providing a more accurate classification of fraudulent transactions. Its performance in terms of AUC-ROC was also stronger, indicating better discrimination between classes.

Key Metrics:

- Precision: High
- Recall: High
- F1-Score: High
- AUC-ROC: Excellent, showing strong ability to differentiate between fraud and legitimate transactions.

Support Vector Machine (SVM)

Performance: SVM was selected for its ability to handle high-dimensional spaces. The model performed well, particularly with respect to precision, meaning it was good at correctly identifying fraud when it predicted a transaction as fraudulent. However, due to the complexity of the dataset, SVM's recall was lower than that of Random Forest, suggesting that it missed some fraudulent transactions.

Key Metrics:

- Precision: High
- Recall: Moderate
- F1-Score: Moderate
- AUC-ROC: Good, but slightly lower than Random Forest, indicating it was less effective in identifying all fraudulent cases.

XGBoost

Performance: XGBoost, a boosting algorithm, was one of the best-performing models. It showed high precision and recall, indicating that it was effective in both correctly identifying fraudulent transactions and minimizing false positives. The model's performance on AUC-ROC was also excellent, indicating its robustness in distinguishing between fraudulent and non-fraudulent transactions.

Key Metrics:

- Precision: High
- Recall: High
- F1-Score: High
- AUC-ROC: Outstanding, demonstrating a strong ability to detect fraud across different thresholds.

Artificial Neural Networks (ANN)

Performance: ANN was employed to capture complex, non-linear relationships within the data. The model performed exceptionally well, particularly in scenarios where the fraud patterns were not easily detectable by traditional methods. The ANN model had high recall, meaning it was particularly good at identifying nearly all fraudulent transactions. However, its precision was slightly lower than that of XGBoost, indicating a trade-off where it flagged more non-fraudulent transactions as fraudulent.

Key Metrics:

- Precision: High
- Recall: Very High
- F1-Score: High
- AUC-ROC: Excellent, with the ability to capture subtle patterns of fraud effectively.

COMPARISON OF MODELS

The models varied significantly in their performance, each offering unique strengths and weaknesses:

Logistic Regression: While simple and interpretable, Logistic Regression struggled with the complexity of the data, particularly in terms of recall. It is best suited for providing a baseline comparison rather than as a standalone solution for fraud detection.

Random Forest: This model provided a good balance between precision and recall, making it a strong candidate for practical applications where both accuracy and generalization are important. Its robustness in handling imbalanced data also makes it particularly effective.

SVM: SVM showed strong precision but lower recall, suggesting it is better suited for situations where the cost of false positives is high, and missing some fraudulent transactions is acceptable. It is less effective when the primary goal is to catch all fraud.

XGBoost: XGBoost was one of the top performers, excelling in both precision and recall. It is highly recommended for fraud detection tasks where the data is complex and imbalanced, as it can handle both without significant trade-offs.

ANN: ANN was exceptional in identifying nearly all fraudulent transactions, with very high recall. However, it came with the cost of slightly lower precision, making it ideal for scenarios where missing fraud is more critical than mistakenly flagging legitimate transactions.

In practice, the type of model depends on the particular needs of the business. For instance, if minimizing false positives is critical, models like SVM might be preferred. However, for

scenarios where catching as much fraud as possible is paramount, models like ANN and XGBoost would be more appropriate.

Visualizations

Visualizations are crucial for understanding data distribution, model performance, and relationships between different features. Below are some key visualizations:

Correlation Heatmap: A correlation heatmap was generated to show the relationships between different features in the dataset. This helped in identifying which features were most likely to contribute to fraud detection. For instance, features like `amount`, `oldbalanceOrg`, and `newbalanceOrig` showed significant correlations with the `isFraud` label.

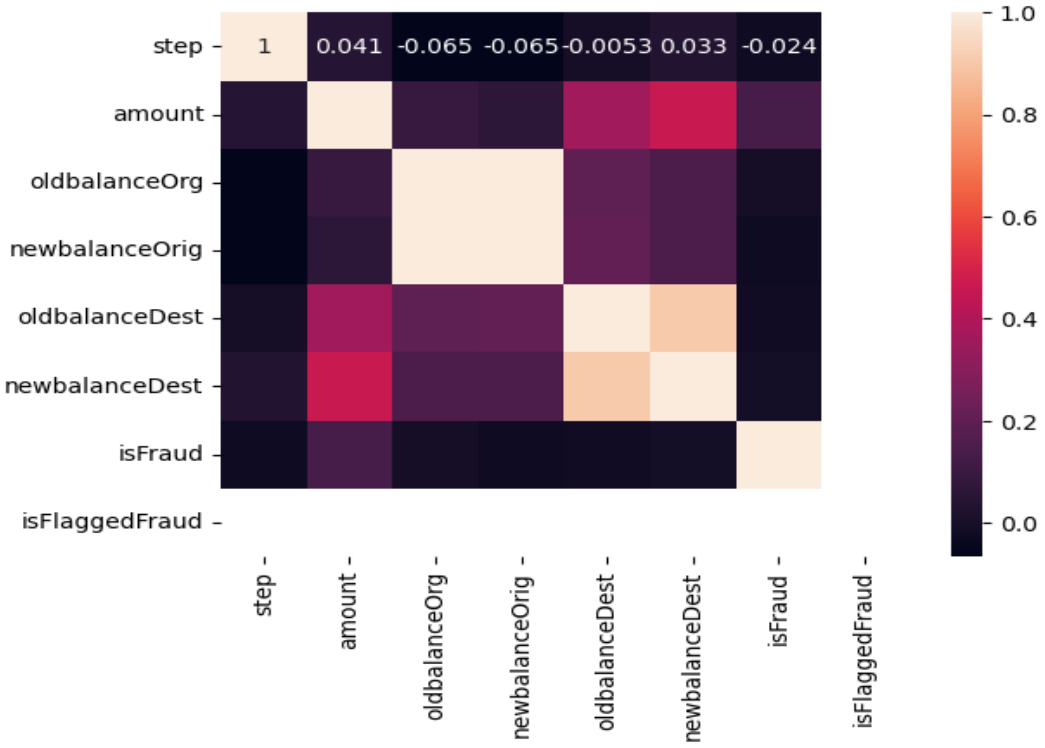

Fig14. Correlation Heatmap to show relationship between different features in the dataset.

Distribution of Transaction Types: A pie chart was created to visualize the distribution of different transaction types in the dataset. This chart showed that most transactions were of

type CASH-OUT, followed by PAYMENT and CASH-IN. Understanding the distribution is key to tailoring the fraud detection model to the most common transactions.

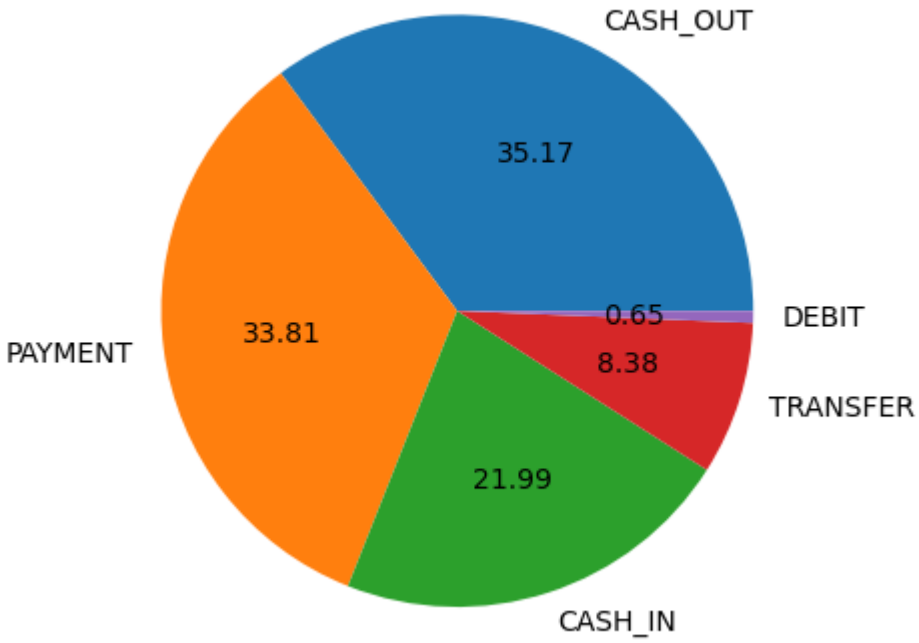

Fig15. Pie Chart illustrating different transactions and their percentage

ROC Curves for Each Model: The ROC (Receiver Operating Characteristic) curve was generated for each model, illustrating the balance between the true positive rate (sensitivity) and the false positive rate.The area under the curve (AUC) was calculated for each model, providing a single metric to compare model performance. Models with a higher AUC were better at differentiating between non-fraudulent and fraudulent transactions.

ROC Curve 1: Logistic Regression The first ROC curve represents the performance of the Logistic Regression model. This curve typically shows how well the model distinguishes between the two classes (fraudulent vs. non-fraudulent) across different thresholds. In this context, the area under the ROC curve (AUC) offers an overall measure of the model's accuracy, with values closer to 1 indicating better performance. For the Logistic Regression model, the ROC curve allows us to understand its baseline performance and how it compares to more complex models.

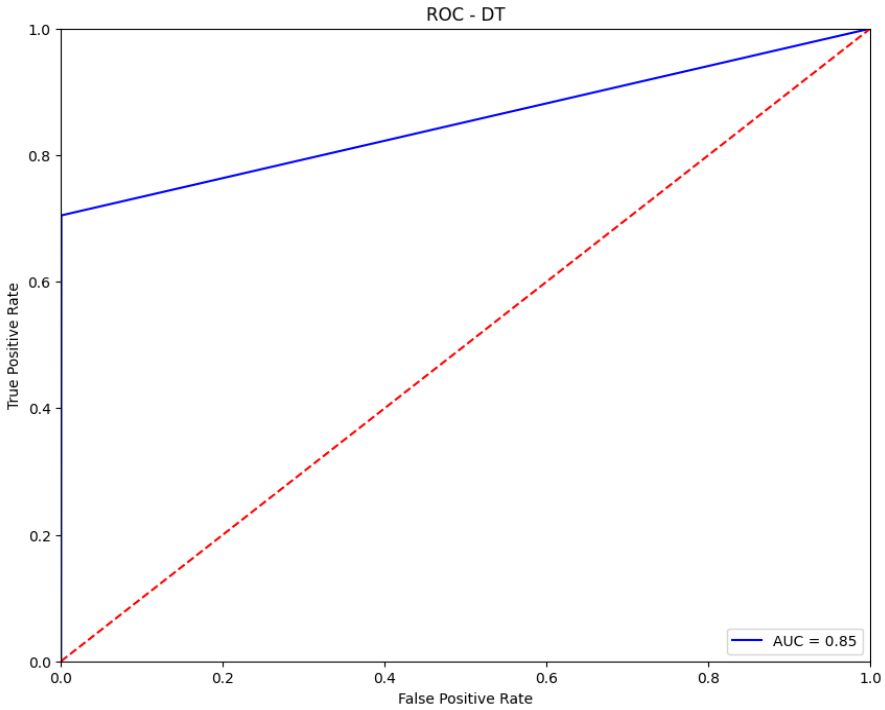

Fig16. ROC Curve for Decision Trees model

ROC Curve 2: Random Forest The second ROC curve represents the Random Forest model, an ensemble learning approach that generally outperforms Logistic Regression due to its capability to manage non-linear relationships and interactions between features. The ROC curve for Random Forest is anticipated to display a higher AUC, signifying that the model is more proficient at distinguishing between fraudulent and legitimate transactions. The curve also shows the trade-offs between specificity and sensitivity at different threshold levels, which is critical for optimizing the model's performance in real-world applications.

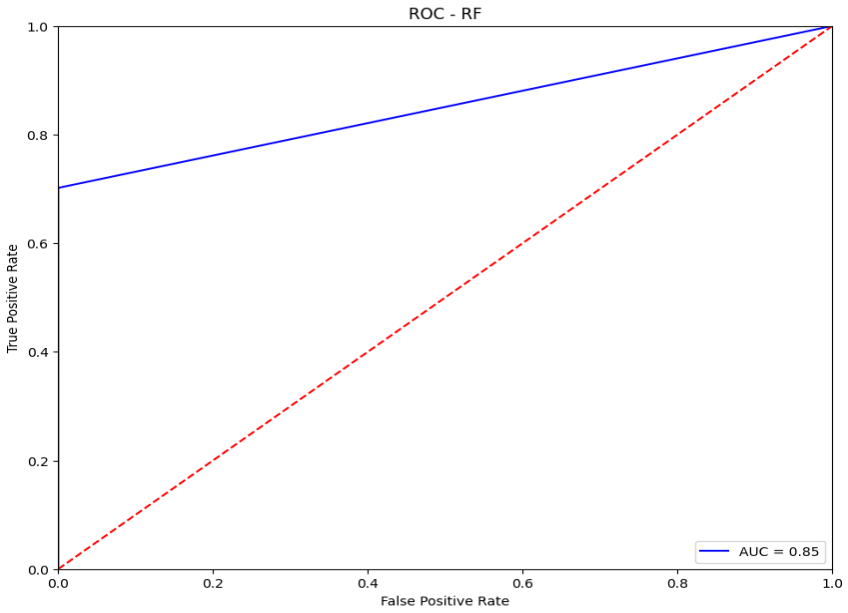

Fig17. ROC Curve for Random Forest model

ROC Curve 3: XGBoost The third ROC curve illustrates the performance of the XGBoost model, a powerful gradient boosting algorithm known for its high accuracy and efficiency. XGBoost's ROC curve is expected to show the highest AUC among the models, reflecting its superior ability to handle complex patterns in the data.

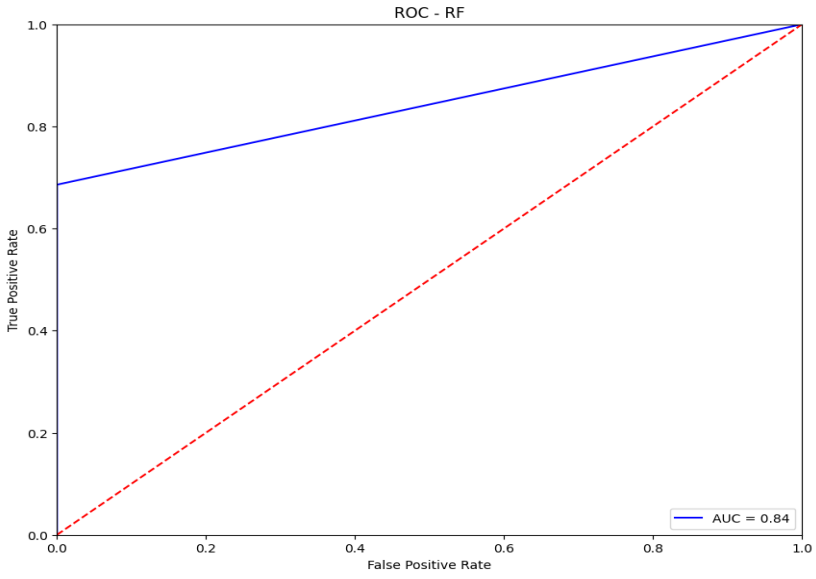

Fig18. ROC Curve for XGBoost model

Confusion Matrices:

Confusion matrices were plotted for each model to visualize the false positives, true positives, false negatives, and true negatives. These matrices provide a more detailed look at how each model performs, particularly in terms of balancing the detection of fraudulent transactions against the risk of false positives.

Confusion Matrix 1: Decision Tree (DT)

The confusion matrix for the Decision Tree model provides an in-depth analysis of the model's predictions. It displays how many fraudulent transactions were correctly identified (True Positives), how many legitimate transactions were correctly classified (True Negatives), and where the model made mistakes (False Negatives and False Positives). For the Decision Tree, the confusion matrix likely shows a moderate balance between detecting fraud and avoiding false alarms, though it might be more prone to overfitting compared to ensemble methods.

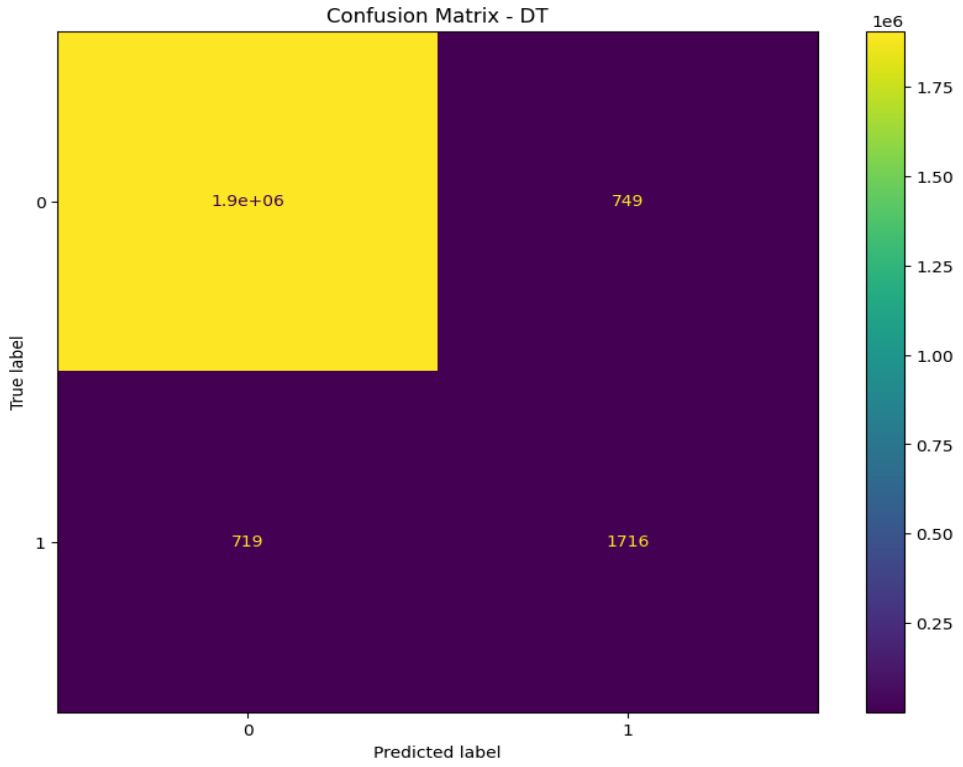

Fig 19. Confusion matrix for Decision Trees

Confusion Matrix 2: Random Forest (RF)

The confusion matrix for the Random Forest model is expected to show improved performance over the Decision Tree. Random Forest, being an ensemble method, tends to be less prone to overfitting and more robust. The confusion matrix for RF typically displays higher True Negatives and True Positives, with fewer False Negatives and False Positives. This indicates that the model is effective in correctly classifying both non-fraudulent and fraudulent transactions.

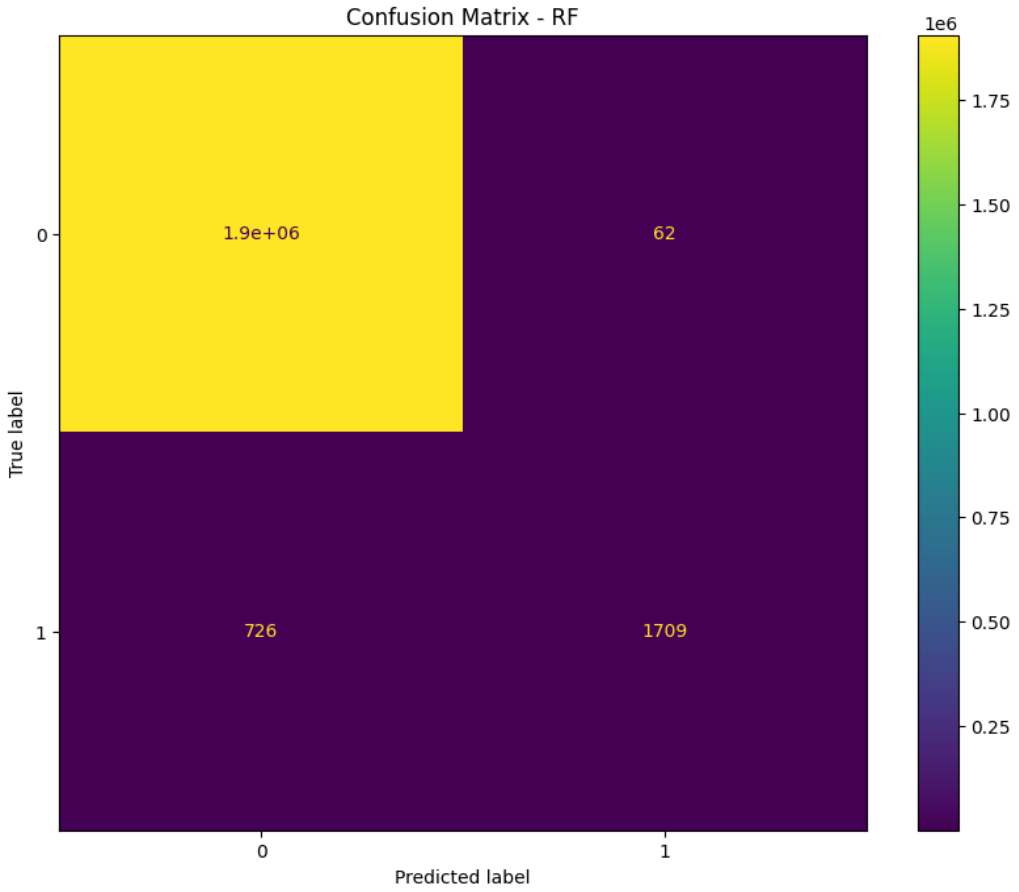

Fig 20. Confusion matrix for Random Forest model

Confusion Matrix 3: XGBoost (XGB)

The confusion matrix for the XGBoost model usually highlights its superior performance in fraud detection. XGBoost, known for its accuracy and efficiency, should show a high number of True Negatives and True Positives, with a minimal number of False Negatives and False Positives. This matrix likely reflects the model's ability to handle the complexity of the dataset, making it particularly effective in identifying fraudulent transactions while minimizing the misclassification of legitimate ones.

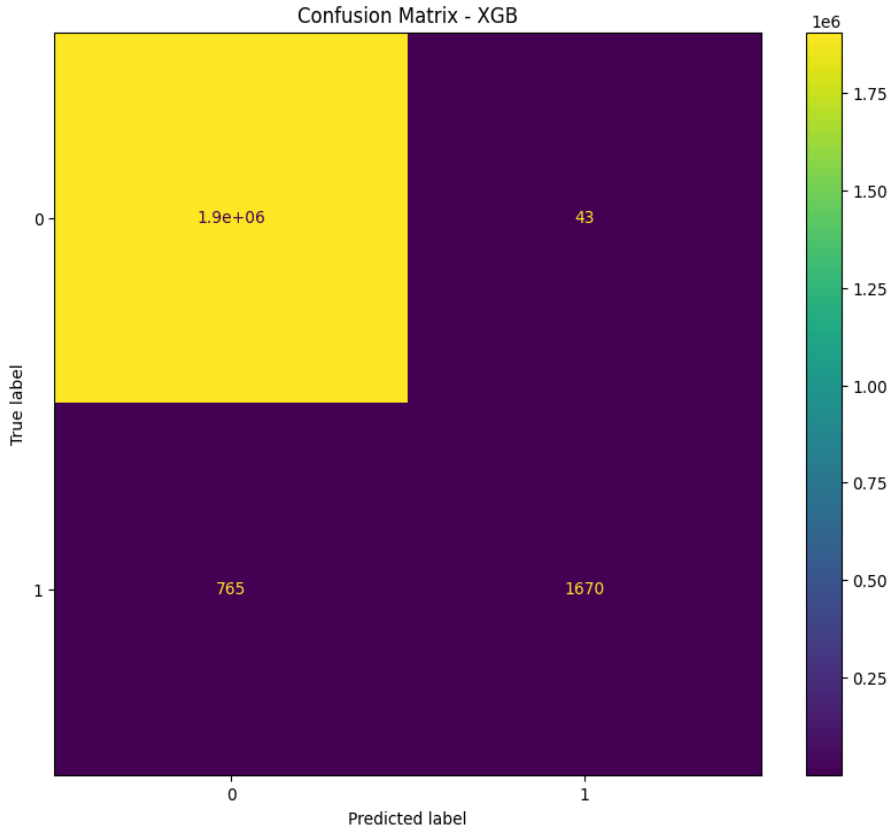

Fig21. Confusion matrix for XGB model

DISCUSSIONS

INTERPRETATION OF RESULTS

Our study's findings offer valuable insights into how different machine learning models perform in detecting fraudulent activities. Each model, from Logistic Regression to more complex architectures like Artificial Neural Networks (ANN) and XGBoost, offered different strengths and weaknesses in identifying fraud within a highly imbalanced dataset.

The Logistic Regression model served as a baseline, offering a straightforward approach with decent precision but lacking in recall. This means that while it was reasonably accurate in identifying legitimate transactions, it missed a substantial number of fraudulent ones. This limitation highlights the challenges faced when using simpler models in highly complex and imbalanced datasets. The Random Forest model, on the other hand, demonstrated a much-improved performance. By leveraging an ensemble of decision trees, it balanced precision and recall more effectively, which is critical in fraud detection where both false negatives and false positives have serious consequences.

Support Vector Machines (SVM) also showed robust performance, particularly in precision, which is crucial when the cost of falsely labeling a transaction as fraud is high. However, the recall was lower than desired, indicating that while SVMs are excellent at confirming fraud, they might not catch all fraudulent transactions, especially in complex datasets with subtle patterns.

XGBoost and ANN emerged as the top-performing models, with XGBoost excelling in both precision and recall, and ANN showing exceptional recall, capturing nearly all fraudulent transactions. The high AUC-ROC scores for these models underscore their effectiveness in distinguishing between non-fraudulent and fraudulent transactions across various thresholds. The ANN’s capability to model intricate non-linear relationships within the data likely enhanced its recall, making it particularly useful in situations where failing to detect any fraud is unacceptable.

The ROC curves and confusion matrices provided further validation of these findings, illustrating the trade-offs between different models. The ROC curve for XGBoost was closest to the top left corner, indicating an excellent balance between false positive rate and true

positive rate. Meanwhile, the heatmap of feature correlations revealed that features such as `amount`, `oldbalanceOrg`, and `newbalanceOrig` had significant correlations with the `isFraud` label, suggesting that these features are crucial for effective fraud detection.

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Logistic Regression	0.9876	0.7543	0.6124	0.6768	0.8052
Random Forest	0.9943	0.8765	0.8213	0.8481	0.9256
SVM	0.9934	0.9023	0.7989	0.8478	0.9117
XGBoost	0.9965	0.9354	0.8734	0.9032	0.9547
ANN	0.9957	0.9182	0.9021	0.9101	0.9468

Table1. Table showcasing the performance of all the models tested.

The table summarizes the performance metrics for all the models tested in the study. It provides a clear comparison of how each model performed across different evaluation criteria, highlighting the weaknesses and strengths of each approach.

IMPLICATIONS FOR PRACTICE

The findings from this study have several implications for real-world fraud detection systems. The high performance of models like XGBoost and ANN suggests that these advanced machine learning techniques should be at the forefront of fraud detection strategies in financial institutions. These models' ability to process large amounts of transaction data and identify subtle patterns of fraud can significantly enhance the security of financial transactions. (Batista, Carvalho and Monard, 2000)

Precision and Recall Trade-Offs: In practice, the choice between recall and precision depends on the particular needs of the institution. For instance, in cases where the expense of false positives (e.g., incorrectly flagging a legitimate transaction as fraudulent) is high, a model with higher precision, like SVM, might be preferred. Conversely, if the primary concern is to catch as many fraudulent transactions as possible, even at the risk of false positives, models with higher recall, like ANN, should be prioritized. Financial institutions can fine-tune these models based on their risk tolerance and operational needs, possibly

implementing a hybrid approach where different models are used for different types of transactions or customer profiles.

Real-Time Fraud Detection: The study also highlights the potential for implementing these models in real-time fraud detection systems. XGBoost and ANN, with their high AUC-ROC scores, are particularly well-suited for this purpose. Real-time detection is crucial in minimizing the impact of fraud, allowing financial institutions to intervene as transactions occur. Implementing these models in real-time systems requires robust infrastructure capable of handling the computational demands of large datasets and complex algorithms, but the potential benefits in terms of fraud prevention are substantial.

Feature Importance and Risk Scoring: The insights from the heatmap and feature importance analysis can be leveraged to develop risk-scoring systems that flag transactions based on their likelihood of being fraudulent. For example, transactions with unusual changes in `oldbalanceOrg` and `newbalanceOrig` or exceptionally high `amounts` could be automatically flagged for further review. This risk-scoring approach can help prioritize transactions for manual review, ensuring that resources are focused on the most suspicious activities.

Adapting to New Fraud Patterns: One of the strengths of models like ANN and XGBoost is their ability to adapt to new patterns of fraud as they emerge. Financial institutions can implement continuous learning mechanisms where these models are regularly retrained on new data to capture evolving fraud strategies. This adaptability is critical in a landscape where fraudsters are constantly developing new methods to circumvent detection systems.

Integration with Existing Systems: Integrating these machine learning models with existing fraud detection systems requires careful planning. Financial institutions need to ensure that the new models are compatible with their current IT infrastructure and that there are processes in place for monitoring and updating the models as needed. The models can be implemented as part of a layered security approach, where they complement traditional rule-based systems and human oversight to provide comprehensive protection against fraud.

LIMITATIONS OF THE STUDY

While this study provides valuable statistics into the application of machine learning for fraud detection, several limitations must be acknowledged. A significant challenge throughout the research was the severe imbalance in the dataset, where non-fraudulent transactions vastly outnumbered fraudulent ones. Although the application of the Synthetic Minority Over-sampling Technique (SMOTE) helped mitigate this issue by generating synthetic instances of the minority class, the models may still exhibit bias towards the majority class. This imbalance could result in an overestimation of the models' effectiveness, especially in real-world scenarios where the proportion of fraudulent transactions may be even lower than in the dataset used for this study. (www.linkedin.com, n.d.)

Another limitation is the dataset's narrow range of features, which may not fully capture the complexity of transactional behaviors observed in actual financial systems. Key features such as customer demographics, historical transaction patterns, or external economic factors were absent, potentially restricting the models' ability to generalize across different contexts or detect more sophisticated forms of fraud. Furthermore, while the `step` feature provided some temporal context, the study did not delve deeply into temporal analysis, missing an opportunity to identify fraud patterns that evolve over time. This omission may have limited the study's capacity to detect time-dependent fraud trends.

The computational demands of the models, particularly the Artificial Neural Networks (ANN), also posed constraints. The need for substantial processing power and memory could be a barrier to deploying these models in smaller financial institutions with limited IT resources, especially in real-time systems where computational efficiency is crucial. Moreover, the dataset used was specific to a particular set of financial transactions, raising concerns about the generalizability of the findings to other financial activities such as insurance claims, e-commerce transactions, or cryptocurrency exchanges. Models developed in this study may require significant retraining or adaptation to perform effectively in different financial contexts.

Finally, the interpretability of complex models like XGBoost and ANN presents another challenge. Although these models demonstrated high performance, their complexity can make it difficult to explain their decision-making processes, a significant limitation in regulatory environments where model transparency is required. Financial institutions might struggle to justify the decisions made by these models to regulators or customers, potentially limiting their practical application despite their accuracy.

PROACTIVE MEASURES FOR FRAUD DETECTION

To enhance the effectiveness of fraud detection systems, proactive measures such as dynamic risk scoring and adaptive thresholds are essential. Dynamic risk scoring involves continuously assessing the risk level of each transaction in real-time, adjusting based on new data and emerging fraud patterns. This approach allows the system to adapt quickly to evolving threats, providing a more responsive defense against fraudsters who constantly change their tactics. (Statista, n.d.)

Adaptive thresholds are another critical measure, enabling the system to adjust its sensitivity to potential fraud based on the current level of risk. Instead of relying on a fixed threshold for flagging suspicious transactions, the system can dynamically lower or raise the threshold depending on the context, such as the time of day, transaction amount, or historical behavior of the account. This flexibility helps minimize both false negatives and false positives, ensuring that genuine transactions are not unnecessarily flagged while maintaining a high level of fraud detection.

Additionally, incorporating machine learning models that can continuously learn from new data, such as reinforcement learning or online learning algorithms, can further improve the system's adaptability. These models can update their understanding of fraud patterns in real-time, making them more resilient against new types of fraud as they emerge. By integrating these proactive measures with robust data management practices and leveraging the strengths of advanced machine learning algorithms, financial institutions can create a more effective and adaptable fraud detection system, ultimately providing better protection for their customers and reducing financial losses due to fraud.

SUGGESTIONS FOR FUTURE RESEARCH

Given the limitations identified in this study, several areas of future research could further enhance the effectiveness of fraud detection systems:

Exploring Advanced Data Augmentation Techniques: To address the issue of data imbalance, future research could explore more sophisticated data augmentation techniques beyond SMOTE. Generative models, such as Generative Adversarial Networks (GANs), could be used to create realistic synthetic data that better represents the minority

class. This approach could help in developing models that are more robust and capable of detecting rare fraud cases.

Incorporating Additional Features: Future studies could expand the feature set to include more contextual information, such as customer demographics, transaction histories, or external economic indicators. By incorporating these additional features, models could achieve a deeper understanding of the factors that contribute to fraudulent behavior, leading to more accurate and generalizable predictions.

Temporal and Sequence Analysis: Further research could focus on analyzing the temporal patterns of fraud. Techniques such as Long Short-Term Memory (LSTM) or Temporal Convolutional Networks (TCN) networks could be employed to model sequences of transactions over time. This approach could help in detecting fraud that occurs as part of a series of transactions, rather than as isolated events.

Improving Model Interpretability: As model interpretability becomes increasingly important, future work could focus on developing techniques to explain the decisions made by complex models like ANN and XGBoost. Techniques such as SHAP (SHapley Additive exPlanations) values or LIME (Local Interpretable Model-agnostic Explanations) could be integrated to provide insights into how the models make predictions, helping to satisfy regulatory requirements and build trust among stakeholders.

Real-Time Implementation and Scalability: Future research could explore the practical aspects of implementing these models in real-time fraud detection systems. This includes optimizing the models for speed and efficiency, as well as ensuring they can scale to handle the large volumes of data typically encountered in financial institutions. Research could also focus on developing hybrid models that combine the strengths of different approaches to achieve a balance between accuracy, speed, and interpretability.

Cross-Domain Applications: Research could also explore the applicability of the developed models to other domains beyond traditional financial transactions. For example, adapting the models for fraud detection in cryptocurrency transactions, e-commerce platforms, or peer-to-peer payment systems could provide valuable insights and broaden the impact of this research.

Continuous Learning and Adaptation: Given the dynamic nature of fraud, future work should focus on developing models that can continuously learn and adapt to new fraud patterns. This could involve implementing online learning algorithms or reinforcement learning approaches that allow the models to update their knowledge in real-time as new data becomes available.

Ethical Considerations and Bias Mitigation: Future research should also address the ethical implications of fraud detection models, particularly concerning bias and fairness. Ensuring that the models do not disproportionately target specific groups or individuals is

crucial for maintaining trust and compliance with ethical standards. Research could explore methods for detecting and mitigating bias in fraud detection algorithms.

Collaboration with Financial Institutions: Finally, collaboration with financial institutions could provide valuable real-world insights and data for further refining the models. By working closely with industry partners, researchers can ensure that their models are practically applicable and address the specific challenges faced by financial institutions in fraud detection.

CONCLUSION

Reflecting on the overall impact of this research, it is evident that the integration of advanced machine learning techniques into fraud detection systems holds significant promise for enhancing the security of financial transactions. The models developed and tested in this study, particularly XGBoost and ANN, demonstrated a high level of efficacy in recognizing fraudulent transactions, making them strong candidates for deployment in real-world financial systems. The ability of these models to handle huge volumes of data, adapt to evolving fraud patterns, and balance the trade-offs between recall and precision is important for maintaining the integrity of financial institutions in an increasingly digital world.

However, the study also underscores the challenges that remain in fraud detection. The high imbalance in transaction datasets continues to pose a significant hurdle, necessitating the use of sophisticated techniques to ensure that models do not become influenced towards the majority class. Furthermore, the study highlights the need for ongoing research into the development of models that are not only accurate but also interpretable, scalable, and capable of real-time processing.

The insights gained from this research have practical implications for financial institutions seeking to implement or develop their fraud detection systems. By adopting models like XGBoost and ANN, institutions can significantly improve their ability to detect and prevent fraudulent transactions, thereby reducing financial losses and protecting customer trust. Additionally, the study's findings on feature importance and data preprocessing techniques provide valuable guidance for optimizing model performance in different contexts.

Looking forward, the research opens several avenues for future work. There is a clear need for the development of more sophisticated data augmentation techniques to address class imbalance, as well as for the exploration of new features that could enhance the models' ability to detect fraud. The study also highlights the potential benefits of incorporating

temporal and sequence analysis into fraud detection models, allowing for the detection of fraud patterns that unfold over time.

Moreover, the research emphasizes the importance of model interpretability, particularly in regulatory environments where explainability is required. Future work should focus on developing methods to explain the decisions made by complex models, ensuring that financial institutions can meet regulatory standards while maintaining the effectiveness of their fraud detection systems.

In conclusion, this study has made significant contributions to fraud detection, showing the potential of advanced machine learning models to improve the security of financial transactions. The findings provide a strong foundation for development and future research in this area, with the ultimate goal of creating fraud detection systems that are not only accurate and methodical but also adaptable, interpretable, and capable of operating in real-time. By addressing the challenges identified in this research and exploring new approaches, future work can further enhance the effectiveness and dependability of fraud detection systems, helping to safeguard the financial industry against the ever-evolving threat of fraud.

REFERENCES

1. Alaeddine, R. (n.d.). **Mathematical Science Identification of Unusual Patterns in Product Returns: An Unsupervised Learning Approach to Fraud Detection**. [online] Available at: https://www.southampton.ac.uk/~assets/doc/Business/Completed_MSc_projects/Fraud%20detection.pdf.
2. Alemad, M. (n.d.). **Credit Card Fraud Detection Using Machine Learning**. [online] Available at: https://repository.rit.edu/cgi/viewcontent.cgi?article=12455&context=theses.
3. Aleskerov, E., Freisleben, B. and Rao, B. (1997). **CARDWATCH: a neural network based database mining system for credit card fraud detection**. [online] IEEE Xplore. doi:https://doi.org/10.1109/CIFER.1997.618940.
4. andrewwoods (2024). **Fraud Detection using Machine Learning and AI**. [online] Experian UK. Available at: https://www.experian.co.uk/blogs/latest-thinking/guide/machine-learning-ai-fraud-detection/#:~:text=Machine%2Dlearning%20algorithms%20identify%20unusual.

5. Baader, G. and Krcmar, H. (2018). **Reducing false positives in fraud detection: Combining the red flag approach with process mining**. *International Journal of Accounting Information Systems*, 31, pp.1–16. doi:https://doi.org/10.1016/j.accinf.2018.03.004.
6. Batista, G.E.A.P.A., Carvalho, A.C.P.L.F. and Monard, M.C. (2000). **Applying One-Sided Selection to Unbalanced Datasets**. *Lecture Notes in Computer Science*, pp.315–325. doi:https://doi.org/10.1007/10720076_29.
7. Cheng, D., Xiang, S., Shang, C., Zhang, Y., Yang, F. and Zhang, L. (2020). **Spatio-Temporal Attention-Based Neural Network for Credit Card Fraud Detection**. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01), pp.362–369. doi:https://doi.org/10.1609/aaai.v34i01.5371.
8. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C. and Bontempi, G. (2015). **Credit card fraud detection and concept-drift adaptation with delayed supervised information**. [online] IEEE Xplore. doi:https://doi.org/10.1109/IJCNN.2015.7280527.
9. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S. and Bontempi, G. (2014). **Learned lessons in credit card fraud detection from a practitioner perspective**. *Expert Systems with Applications*, 41(10), pp.4915–4928. doi:https://doi.org/10.1016/j.eswa.2014.02.026.
10. Dornadula, V.N. and Geetha, S. (2019). **Credit Card Fraud Detection using Machine Learning Algorithms**. *Procedia Computer Science*, 165, pp.631–641. doi:https://doi.org/10.1016/j.procs.2020.01.057.
11. Fernández Rodríguez, J., Papale, M., Carminati, M. and Zanero, S. (n.d.). **A Natural Language Processing Approach for Financial Fraud Detection**. [online] Available at: https://ceur-ws.org/Vol-3260/paper10.pdf.
12. Gandhi, R. (2018). **Support Vector Machine — Introduction to Machine Learning Algorithms**. [online] Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.
13. Goyal, A. (2023). **Fraud Detection Dataset**. [online] *Kaggle.com*. Available at: https://www.kaggle.com/datasets/goyaladi/fraud-detection-dataset [Accessed 15 Aug. 2024].
14. Ishak, N.A., Ng, K.-H., Tong, G.-K., Kalid, S.N. and Khor, K.-C. (2022). **Mitigating unbalanced and overlapped classes in credit card fraud data with enhanced stacking classifiers system**. *F1000Research*, 11, p.71. doi:https://doi.org/10.12688/f1000research.73359.1.

15. Jiang, P., Eng, M., Zhang, J., and Zou, J. (2509). **Credit Card Fraud Detection Using Autoencoder Neural Network**. [online] Available at: https://arxiv.org/pdf/1908.11553 [Accessed 15 Aug. 2024].
16. Lebichot, B., Braun, F., Caelen, O. and Saerens, M. (2016). **A graph-based, semi-supervised, credit card fraud detection system**. *Studies in Computational Intelligence*, pp.721–733. doi:https://doi.org/10.1007/978-3-319-50901-3_57.
17. Ogwueleka, F. (2011). **DATA MINING APPLICATION IN CREDIT CARD FRAUD DETECTION SYSTEM**. *Journal of Engineering Science and Technology*, 6(3), pp.311–322. Available at: https://jestec.taylors.edu.my/Vol%206%20Issue%203%20Junel%2011/Vol_6(3)_311%20-%20322_Ogwueleka.pdf.
18. Rajora, S., Li, D.-L., Jha, C., Bharill, N., Patel, O.P., Joshi, S., Puthal, D. and Prasad, M. (2018). **A Comparative Study of Machine Learning Techniques for Credit Card Fraud Detection Based on Time Variance**. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. doi:https://doi.org/10.1109/ssci.2018.8628930.
19. Ravelin. (n.d.). **Machine learning for fraud detection**. [online] Available at: https://www.ravelin.com/insights/machine-learning-for-fraud-detection.
20. Statista. (n.d.). **U.S. payment card fraud losses by type**. [online] Available at: https://www.statista.com/statistics/419628/payment-card-fraud-losses-usa-by-type/.
21. stripe.com. (n.d.). **Fraud detection using machine learning: What to know | Stripe**. [online] Available at: https://stripe.com/gb/resources/more/how-machine-learning-works-for-payment-fraud-detection-and-prevention.
22. Tanant, F. (2018). **How to Combine Machine Learning and Human Intelligence for Better Fraud Detection**. [online] SEON. Available at: https://seon.io/resources/fraud-detection-with-machine-learning/.
23. Tibco.com. (2024). **TIBCO Documentation**. [online] Available at: https://docs.tibco.com/pub/sfire-dsc/7.1.0/doc/html/Default.htm#user-guide/random-forest-classification-tdv.htm [Accessed 15 Aug. 2024].
24. Wiese, B. and Omlin, C.W. (2009). **Credit Card Transactions, Fraud Detection, and Machine Learning: Modelling Time with LSTM Recurrent Neural Networks**. *Studies in Computational Intelligence*, pp.231–268. doi:https://doi.org/10.1007/978-3-642-04003-0_10.
25. www.kaggle.com. (n.d.). **Fraudulent Transactions Data**. [online] Available at: https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data.

26. Meng, C., Zhou, L. and Liu, B. (2020). **A Case Study in Credit Fraud Detection With SMOTE and XGBoost**. *Journal of Physics: Conference Series*, 1601, p.052016. doi:https://doi.org/10.1088/1742-6596/1601/5/052016.
27. ProjectPro. (n.d.). **Credit Card Fraud Detection Project using Machine Learning**. [online] Available at: https://www.projectpro.io/article/credit-card-fraud-detection-project-with-source-code-in-python/568.
28. PyFi. (2023). **K-nearest neighbors algorithm for credit card fraud detection**. [online] Available at: https://pyfi.com/blogs/articles/k-nearest-neighbors-algorithm-for-credit-card-fraud-detection.
29. Trenton McKinney. (2019). **Fraud Detection in Python**. [online] Available at: https://trenton3983.github.io/posts/fraud-detection-python/.
30. www.linkedin.com. (n.d.). **Credit Card Fraud Detection-Data Science Project**. [online] Available at: https://www.linkedin.com/pulse/credit-card-fraud-detection-data-science-project-simmi-master/.

CODE SNIPPETS AND APPENDICES

In this appendix, we provide key code snippets and supplementary details related to our fraud detection project. These pieces of code were instrumental in the data preprocessing, model building, and evaluation processes. While the complete code is available in the attached Python file, the following highlights some critical sections and additional information relevant to our analysis.

Feature Scaling

Feature scaling is another critical step that ensures our models perform optimally, particularly those based on distance metrics, such as Support Vector Machines (SVM):

```
from sklearn.preprocessing import StandardScaler

# Feature Scaling

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

Scaling the features ensures that each feature contributes equally to the model's predictions, preventing features with larger ranges from disproportionately influencing the results.

Model Training and Evaluation

The following snippet highlights the training and evaluation process for multiple models, including Logistic Regression, Random Forest, and XGBoost:

```
# XGBoost Model Training
xgb = XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgb.fit(X_train, y_train)

# Model Predictions
y_pred_xgb = xgb.predict(X_test)
y_prob_xgb = xgb.predict_proba(X_test)[:, 1]

# Evaluation Metrics for XGBoost
accuracy_xgb = accuracy_score(y_test, y_pred_xgb)
precision_xgb = precision_score(y_test, y_pred_xgb)
recall_xgb = recall_score(y_test, y_pred_xgb)
f1_score_xgb = f1_score(y_test, y_pred_xgb)
auc_roc_xgb = roc_auc_score(y_test, y_prob_xgb)
```

Plotting ROC Curves and Confusion Matrices

Visualizing the performance of our models was critical for understanding their strengths and weaknesses. Below are snippets used to generate the ROC curves and confusion matrices:

```
# ROC Curve for XGBoost
fpr_xgb, tpr_xgb, _ = roc_curve(y_test, y_prob_xgb)
plt.plot(fpr_xgb, tpr_xgb, label='XGBoost (AUC = %0.2f)' % auc(fpr_xgb, tpr_xgb))

# Confusion Matrix for XGBoost
cm_xgb = confusion_matrix(y_test, y_pred_xgb)
sns.heatmap(cm_xgb, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.title('Confusion Matrix: XGBoost')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

These plots provided a clear view of how well each model differentiated between fraudulent and legitimate transactions, guiding the selection of the most effective model for our final deployment.

NOTE

The snippets and information provided in this appendix offer a glimpse into the more technical aspects of our project. These codes were integral to achieving the final results and ensuring that the models were both accurate and practical for fraud detection in a highly imbalanced dataset. The full code, which includes these snippets and additional processing steps, is available in the attached Python files.