# INTRODUCTION

## Background

**Understanding Financial Fraud and its Impact**

In today’s fast-paced world, financial transactions are happening continuously, every millisecond. Most of the financial institutions are moving into digitalization in order to supply the emerging demand arising from consumers globally. With the significant growth of the industry and improved digitalization, there has been a corresponding increase in scammers and fraudsters. Consequently, fraudulent activities have become a major concern for businesses, banks and consumers from around the world as such activities are becoming both more sophisticated and more frequent.

Financial fraud can happen in many forms. Credit card frauds, insurance frauds, identity theft and money laundering are some of the most prominent ways of fraud in the present society. The implications of such transactions are affecting not only individuals and businesses but also the stability of entire financial systems. The global cost of financial fraud is staggering, with estimates running into trillions of dollars annually, which in turn impacts consumer trust, raises operational costs for businesses, and undermines economic growth (ACFE, 2022)

**Traditional Approaches to Financial Fraud Detection**

Given the significant impact of financial fraud, detecting and preventing such activities has become a critical area of focus. Over the years, various machine learning models have been deployed to detect fraudulent activities by analyzing patterns within large and complex datasets. Some of the most commonly used models include:

**Random Forest:** This ensemble learning method constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. It is known for its robustness and high accuracy, particularly in handling large datasets with many features (Breiman, 2001).

**Neural Networks:** These are inspired by the human brain's structure and function. Neural networks, particularly deep learning models, have shown remarkable performance in identifying complex patterns within data. However, they often require large amounts of labeled data and can be computationally expensive (LeCun, Bengio, & Hinton, 2015).

**XGBoost**: An advanced implementation of gradient-boosted decision trees, XGBoost is designed for speed and performance. It has been widely adopted for its ability to handle imbalanced datasets, which are common in fraud detection scenarios. However, like other tree-based methods, it may struggle with generalization if the fraud patterns are highly dynamic (Chen & Guestrin, 2016).

**Limitations of Traditional Models**

Despite the popularity of these traditional fraud detection models, there are concerns about their effectiveness in identifying new and evolving patterns of fraudulent transactions. These models highly rely on the historical data, rather than identifying evolving patterns of fraudulent transactions. This highlights a raising question about the abilities of these traditional models in detecting emerging fraud trends.

**Introduction to Retrieval-Augmented Generation (RAG)**

In recent years, advancements in the artificial intelligence, particularly in the field of natural language processing (NLP), have introduced few possibilities having the potential to overcome thus limitations. One such advancement is Retrieval Augmented Generation (RAG), a novel approach which combines retrieval-based models with generative models. RAG has shown promising results in different domains such like question answering and text generation, by leveraging the large corpus of relevant information during inference. In RAG, a retrieval mechanism is employed to gather relevant documents or data segments from an external corpus, which are then fed into a generative model (like GPT) to produce a final output (Lewis et al., 2020). This hybrid approach allows the model to generate informed responses based on both the input data and the retrieved information, leading to more accurate and contextually rich predictions.

This research explores the potential of integrating Retrieval-Augmented Generation into traditional fraud detection models to address the limitations of current approaches. The study aims to evaluate the effectiveness of RAG in improving the detection of fraudulent transactions, particularly in scenarios where existing models may fall short, such as in cases of novel or subtle fraud patterns. By enhancing the ability of fraud detection systems to dynamically retrieve and utilize relevant information, this research seeks to contribute to the development of more robust and adaptable solutions for combating financial fraud.

## Problem Statement

Financial fraud continues to be a significant threat to the global economy. While fraudulent transactions are becoming increasingly sophisticated and more frequent, the ability to detect and prevent of thus transactions is becoming further challenging. Traditional models like Random Forest Classification, Neural Network models and even XGBoost are widely equipped around the world in detection of fraudulent activities. However, these models fall short in identifying new or evolving fraud patterns due to their reliance of historical data and incapability of retrieving external, contextually relevant information (Breiman, 2001; LeCun, Bengio & Hinton, 2015; Chen & Guestrin, 2016).

As fraudsters continuously refine their tactics, there is a critical demand for a more advanced model that can dynamically integrate external knowledge and provide context aware analysis. Retrieval Augmented Generation (RAG) provides a promising solution by combining both retrieval based models with generative models. This enhance the detection of complex and emerging fraud patterns. However, the application of RAG in the context of financial fraud detection remains underexplored.

This project aims to address the gap by investigating the potential of RAG in improving the accuracy and effectiveness of financial fraud detection systems. By leveraging the capabilities of RAG, this research seeks to develop a more robust and adaptive approach to combat financial fraud.

## Justification of the Study

Due to prevailing economic issues around the world, financial fraud has increased significantly throughout the world. According to the UK Finance annual fraud report 2024, losses due to fraud in the UK reached £1.17 billion in 2023 (UK Finance, 2024). This implies the current systems in place are not successful enough, demanding a more advanced model which can detect novel and complex patterns of fraud.

Retrieval Augmented Generation (RAG) has shown promising results in the realm of Natural Language Processing and widely used in creating question answering systems. In this research I tried the potential of leveraging the retrieval and generation capabilities of a RAG model and incorporate It into identifying fraudulent transactions.

In the realm of data science and machine learning, usage of embeddings, vector stores and querying is underexplored in the context of non-textual datasets. Hence, there was a urge to utilize RAG model along with embeddings, vector stores, indexing and querying in a structured transactional dataset, in this case the financial fraud transaction dataset. I took this opportunity to investigate this un-explored area.

In this project, I have utilized traditional models like Random Forest, Neural Network and XGBoost to set up a benchmark to compare the performance of RAG integrated model. Also I have tried multiple measuring metrics when it comes to finding the nearest matching transaction based on the similarity score. Additionally, creating a protype model of generating responses based on the user query prompt is also considered, as the need for such model is helpful in detecting fraudulent transactions more effectively.

## Research Questions

This research aims to answer the following questions.

1. What are the capabilities of Retrieval-Augmented Generation (RAG) in identifying fraudulent transactions?
2. How does the effectiveness of Retrieval-Augmented Generation in detecting fraudulent transactions compare to traditional machine learning methods?
3. How can a prototype model be developed using generative AI to retrieve related transactions based on user queries?

## Aims and Objectives

The primary aim of this research is to explore and evaluate the effectiveness of Retrieval Augmented Generation (RAG) in detection of fraudulent transactions within financial dataset.

**Objectives**

**Assess the Capabilities of RAG:** Investigate how Retrieval-Augmented Generation (RAG) can be utilized to identify and analyze fraudulent transactions  
**Compare RAG with Traditional Methods:** Evaluate the performance of RAG in detecting fraudulent transactions relative to established machine learning techniques, such as Random Forest, Neural Networks, and XGBoost, in terms of accuracy, adaptability, and effectiveness.  
**Develop a Prototype Model:** Create and implement a prototype model that leverages generative AI to retrieve and analyze related transactions based on user queries, demonstrating the practical application of RAG in fraud detection.  
**Customize Traditional Models:** Customizing traditional models to capture more insight and increase the accuracy of predictions  
**Investigate how class imbalance effect model performance:** Conducing experiments with imbalanced dataset and oversampled dataset and see how it effects model performance

# REVIEW OF LITERATURE

**Financial Fraud Detection**

Financial fraud is a persistent issue vastly impacting not only the individuals and financial institutions but also the global economy itself. Early detection of fraudulent activities is crucial due to the growing fraud schemes. Various machine learning models have been developed and deployed to address this challenge.

**Traditional models for Fraud Detection**

Traditional models for fraud detection primarily rely on rule-based system and statistical methods. These systems utilize pre-defined set of rules identified by the subject experts to flag suspicious activities. These rules are more often derived from historical data and domain knowledge (Phua, Lee, Smith & Gayler, 2010). Statistical methods, including logistic regression, is also widely used to model the probability of fraud occurrences (Bolton & Hand, 2002). While this model provides a foundation for fraud detection, these models mostly struggle with scalability and adaptability to new and evolving patterns.

**Random Forest**

Random Forest is a robust ensemble learning method widely known for high accuracy in fraud detection models. It operates by constructing multiple decision trees during training and outputting the mode of the classes for classification tasks (Breiman, 2001). Research by Abdallah, Maarof, and Zainal (2016) demonstrated that Random Forest outperforms other algorithms, such as Decision Trees and Logistic Regression, in detecting financial fraud due to its ability to handle large datasets and reduce overfitting.

**Neural Network**

Neural network, especially deep learning models have shown promising results in the context of fraud detection. These models are inspired by the human brain's structure. Neural network model consists of multiple layers which process input data through weighted connections (Goodfellow, Bengio & Courville, 2016). Wei, Hu, and Zhang (2019) applied deep learning techniques to detect fraudulent credit card transactions. It has achieved higher detection rates compared to traditional methods. These models can capture complex patterns and relationship of data. Hence, neural network models can identify subtle and sophisticated fraud schemes.

**XGBoost**

XGBoost is an implementation of gradient-boosted decision trees. This has emerged as a powerful tool for fraud detection due to its efficiency and performance (Chen & Guestrin, 2016). These models particularly capable of handling large scale datasets and provide better generalization. A study by Pourhabibi et al. (2020) highlighted XGBoost's superior accuracy in fraud detection tasks. This ensures the model's ability to focus on hard-to-classify cases by adjusting weights iteratively.

**Class Imbalance and SMOTE**

One of the significant challenges when dealing with financial fraud analysis dataset is the class imbalance problem. The ratio of majority vs minority class is highly different. These imbalances can lead to biased models that favors the major class resulting in poor detection of fraudulent transactions (Jing et al., 2018).

The Synthetic Minority Over-Sampling Technique (SMOTE) is a widely used method to address this issue. SMOTE generates synthetic data for the minority class by interpolating between existing minority instances, thus balancing the dataset (Chawla et al., 2002). In the context of fraud detection SMOTE has shown a significant improvement in the performance of models by providing a more balanced representation of classes. Balancing the weight among the classes is critical for effective model training (Dal Pozzolo et al., 2015).

**Embedding, Vector Stores and Similarity Search**

Embeddings convert categorical variables into continuous vector representation, capturing the semantic relationship among them. This approach is undoubtedly useful in financial data where categorical features like ‘transaction type’ , ‘transaction behavior patterns’ are prevalent (Mikolov et al., 2013).

Vector stores allow for efficient storage and retrieval of these embeddings. These vector stores also allow similarity search on these embeddings. Similarity search is a technique used to find instances in the data according to the query instance provided. In the context of financial fraud this basically mean detecting transactions which are similar to known fraudulent transacitons (Johnson, Douze & Jégou, 2019).

**Sentence Transformers**

Sentence Transformer models are designed to generate semantically meaningful sentence embeddings that can be used for tasks like similarity search, clustering and classification (Reimers & Gurevych, 2019). A study by Wang et al. (2021) demonstrated the effectiveness of Sentence Transformers in financial text classification, which can be modified and applied to fraud detection models.

**Retrieval Augmented Generation (RAG)**

RAG is a relatively recent paradigm in natural language processing (NLP). The primary goal of RAG is to enhance the quality and relevance of generative text by integrating a retrieval mechanism and generative model. Unlike traditional text generation models purely rely on pre-trained data, RAG retrieve relevant information to inform their output. In below sections I will talk about a comprehensive overview of RAG, discussing key research developments, their findings strengths and limitations.

1. The concept of Retrieval Augmented Generation

RAG combines retrieval-based methods with generative models to produce more contextually accurate and informative text. Lewis et al. (2020) introduced the RAG model, which integrates a dense retrieval mechanism with a generative model based on BART (Bidirectional and Auto-Regressive Transformers). The retriever searches an external corpus to find relevant passages. Then generator uses it to generate coherent and context rich response. This dual approach allows RAG to stand out from traditional generative models, which suffers from hallucination and limited factual accuracy (Lewis et al., 2020).

1. Mechanism of RAG

The RAG model consists of two main components: the retriever and the generator. The retriever uses dense passage retriever (DPR) or other embedding based method efficiently search through large datasource. Karpukhin et al. (2020) explored the use of dual encoders for dense retrieval. It demonstrated high accuracy in selecting relevant documents by encoding both queries and documents into a shared vector space. The generator leverages these retrieved documents to produce the final output. Generator models such as BART or GPT to generate coherent and contextually enriched responses (Karpukhin et al., 2020)

A diagram of a computer process

Description automatically generated

Figure1: Overview of Retrieval-Augmented Generation (AWS, 2024)

1. Advancements of Retrieval Mechanisms

Recent research has focused on improving retrieval mechanisms to enhance RAG performance. Qu et al. (2021) proposed an improved retrieval strategy that combines both sparse and dense retrieval techniques to better handle diverse types of queries. This hybrid approach allows for more nuanced retrieval. Also, it balances precision and recall.

1. Enhancements in Generative process

Shuster et al. (2021) explored various ways to incorporate retrieved knowledge into the generation processes. They have tried direct concatenation of retrieved documents and query-focused re-ranking to prioritize more relevant passages.

1. Applications of RAG in NLP Tasks

RAG has been successfully applied in the context of NLP and showcased promising results. In dialogue systems, RAG has improved response relevance and coherence by retrieving pertinent background information to inform the conversation (Shuster et al., 2021)

By thoroughly observing these facts and to fill in the gap of RAG usage in more structured datasets, I have taken the step out to investigate how RAG can be utilized in detecting frauds in a transactional dataset.

**Related work in the area**

The use of Retrieval-Augmented Generation (RAG) in fraud detection is a new and emerging area of research. RAG is widely used in question answering models. Rag retrieves contextual information from a large dataset and uses it to guide predictions, improving detection accuracy.

Lewis et al. (2020) introduced RAG and showcased its effectiveness in tasks requiring extensive knowledge. They found that RAG outperforms traditional transformer models by incorporating a retrieval step which brings external knowledge. While they primary has focused on Natural Language Processing tasks, the principals of RAG too can be applied to fraud detection. It can retrieve past transaction fraud patterns or similar cases, informing the detection model.

RAG can identify complex fraud schemes that traditional models might miss. For example, Khattak et al. (2022) used RAG in a legal question-answering system. They have shown how RAG could generate precise answers by retrieving and integrating relevant cases. Similarly, this can be applied in the domain of fraud detection as well. Integrating RAG into this domain could lead to more nuanced and informed decision making, especially in complex and more sophisticated cases where traditional models might fail.

While the direct application of RAG in the domain of fraud detection is still in early stages, there are lot of potential benefits.

# RESEARCH METHOD

## Data Overview

The dataset used in this project is taken from Kaggle.com repository. According to the repository, the dataset is sourced from BankSim, comprising an aggregated sample of transactional data from a Spanish bank. Dataset consists of 594,643 transactions from unique 4112 users. The target variable(isFraud) identifies fraudulent transactions, while the remaining seven columns include a time step identifier, personal information about the payer such as gender, age group, zip code and transaction details including merchant name, category, and transactional amount. Out of all the transactions in the dataset only 7200 records are identified as fraud, making the dataset highly imbalanced.

A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

## Review of Columns

Original dataset consists of below columns.

|  |  |  |  |
| --- | --- | --- | --- |
| Attribute | Description | Keep? | Explanation |
| Step | Time-step identifier | Yes | Values ranging from 0-180. Assuming a time-step represents a one transactional day. |
| customer | Customer identifier | Yes | Categorical feature. Random unique number to identify customer, ensuring the anonymity |
| age | Age group of customers | Yes | Values ranging from 0-6. |
| gender | Gender of the customer | Yes | Unique values are ‘M’, ‘F’ and ‘U’ |
| zipCodeOri | Zip code of the originated location | No | Removing as the value is static throughout the dataset |
| Merchant | Identification of the merchant | Yes | Categorical feature. Random unique number to identify customer, ensuring the anonymity |
| zipMerchant | Zip code of the merchant location | No | Removing as the value is static throughout the dataset |
| Category | Transaction type | Yes | Categorical feature. Specify for which category the transaction is originated |
| Amount | Transaction amount | Yes | Numerical Feature. |
| Fraud | is Fraud yes or no | Yes | Target variable. If fraud=0 means legitimate transaction. Fraud =1 means transaction is identified as Fraud. |

Before diving into data processing and methodology, I will briefly explain the code structure and integration used in this project. The code is written in Python. I used Google Colab framework, with GitHub used for version control. To streamline the workflow, I introduced a novel approach by directly integrating GitHub with Google Colab. This allowed for seamless communication with GitHub without any hassle. Additionally, I adopted a modular code structure using classes, instances, and functions to enhance readability and minimize redundancy.

## Data Pre-processing, EDA and Feature Engineering

Pre-processing was performed on the dataset to make it more informative and suitable for modeling. This involved several steps: converting categorical data into numerical representations using label encoding, removing duplicate entries, handling missing or unknown values, and eliminating unwanted characters from the dataset. These steps were essential to ensure the dataset was clean and model ready.

Exploratory Data Analysis (EDA) was conducted by visualizing the dataset across various attributes to gain insights into the data's behavior and relationships.

A diagram of a customer age

Description automatically generated

A graph with blue lines

Description automatically generated

A graph with numbers and text

Description automatically generated with medium confidence

A diagram of red and blue dots

Description automatically generated

By observing these plots, I have got a brief understanding of the data set. It is evident that majority of the recorded customers are females and are in age category of 2 and 3. Also the merchant category used in this data set is widely used for transportation. Transaction amount is varying from cents to more than 8000.

I also plotted various graphs to explore the relationships among different variables and their connection to the target variable (isFraud)

By closely observing the plots created against fraud dataset, high number of frauds are observed in customers of 2,3 and 4 age group. Least number of frauds is observed in the 0,6-age group. Comparing the gender of customers marked as fraud, female customers are almost double compared to male. Also, highly flagged merchant category is sports and toys, and it is followed by health. Transactions done in fashion and in tech seems to be more legitimate.

A graph with blue lines

Description automatically generated

A graph of a number of numbers

Description automatically generated with medium confidence

A graph with blue lines

Description automatically generated

A group of graphs with numbers

Description automatically generated with medium confidence

A screenshot of a graph

Description automatically generated

However, upon reviewing the relationship plots, it is evident that most variables do not significantly influence the target variable. Only amount shows a high correlation.

Feature engineering is the process of creating, transformation or selecting relevant features from dataset to improve the performance of the machine learning model. According to Guyon and Elisseeff (2003), feature engineering includes the identification and selection of variables that are most relevant for the model, which can significantly impact its accuracy and efficiency.

I have engineered below additional features to increase the model accuracy.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| encoded\_gender | Encoded from customer gender column. Used label encoding |
| encoded\_category | Encoded from merchant category column. Used label encoding |
| IslargeTransaction | This feature is created to categorize and flag large transactions. I have used a value of 5000 as threshold (changeable). |
| countForCustomerSameTime | This is counter which counts number of transactions done by same customer in a same time step |
| avgTransactionAmount | Calculates and store average transaction amount for each customer |
| transaction\_behavior | Stores a text prompt of behavior of transaction.Used for text embeddings. |

One of the major challenges of the dataset is that it is highly imbalanced.

A blue rectangular bar with red and white text

Description automatically generated

This could lead a model’s prediction’s more biased towards the majority class. Hence. Steps to mitigate the imbalance issue was required. Out of many ways, I have used Synthetic Minority Over-Sampling Technique (SMOTE) to tackle this problem.

A graph with red and blue squares

Description automatically generated

# Trained algorithms to analyze the performance

Before performing the data on RAG model, I have tried the dataset with traditional models to set up a benchmark performance. I have further incorporated customized model optimizations, hype parameter tuning to these traditional models expecting to obtain a better performance accuracy.

## Random Forest Classifier

The Random Forest Classifier is a robust ensemble learning method. It is widely used for classification tasks such as fraud detection. It operates by creating multiple decision trees while training and aggregate their output to make a final decision. This process involves two key techniques: Bagging (Bootstrap Aggregating) and the Random Subspace Method. Bagging creates multiple subsets of the dataset through random sampling. Each sample is used to train a different decision tree. The Random Subspace Method introduces randomness by selecting random subsets of features for each tree. It reduces the correlation among them and enhance overall performance (Breiman, 2001). Each decision tree in the forest is built by recursively splitting the data based on feature values. Final prediction is made based on the majority vote from all trees (Liaw and Wiener, 2002).

### Model Configuration

|  |  |  |
| --- | --- | --- |
| Configuration | Value | Description |
| n\_estimators | 100 | Number of trees in the forest. More the better, however could impact computational cost. |
| max\_depth | None | Maximum death of the tree. I have configured to expand until all leaves are pure |
| min\_samples\_split | 2 | Minimum number of samples required to split an internal node |
| min\_samples\_leaf | 1 | Minimum number of samples at leaf level |

### Model Customizations and Optimizations

1. Handling data imbalance problem by SMOTE oversampling and adjusting the bias towards model.
2. Label encoding done to utilize categorical features in model predictions.
3. Standard scaling to standardize features to have zero mean and unit variance.

## Neural Network Model

Neural Network (NN) models are inspired by the human brain and designed to recognize patterns and make decisions based on input data. They consist of multiples layers connected with neurons. Each connection has an associated weight that adjust while training. The network learns by passing data through these layers. This process enables NN to model complex relationships and perform classification and regression tasks efficiently (Goodfellow, Bengio & Courville, 2016)

## Model Architecture

|  |  |
| --- | --- |
| **Configuration** | **Description** |
| Input layer | Based on the input dataset shape. |
| Dense layer | 1st layer – 64 neurons, Activation - ReLU |
| Dropout layer | Dropout rate - 0.5 |
| 2nd dense layer | 32 neurons, Activation - ReLU |
| Output layer | Single neuron, Activation - Sigmoid |

Model was trained for 10 epochs in 32 batches. Trains and test set ratio was 75% to 25%.

Model uses ‘Adam’ optimizer to adjust the weights of the neurons based on the difference between predicted and actual outcomes. Binary cross-entropy is used as the loss function to measure the error. Additionally, model is customized using early stopping mechanism to halt training if the model’s performance on validation data stops improving, resulting in preventing overfitting.

### Model Customizations and Optimizations

1. Early stopping to prevent overfitting.
2. Adding dropout layers to regularize 50% of neurons to 0 during each training iteration. Expecting model to not rely too much on specific neurons, ultimately reduce overfitting
3. Feature standardization.
4. Label encoding.

## Extreme Gradient Boosting (XGBoost)

XGBoost is an advanced implementation of gradient boosting framework specially designed for its speed and performance (Chen & Guestrin, 2016). The key innovation of XGBoost lies in its use of advanced regularization techniques, parallel computation, and optimization to enhance the predictive performance of the model.

### Model Architecture

The XGBoost classifier is used for its efficiency and effectiveness in handling classification tasks. It builds multiple decision trees to make predictions and aggregates their results to provide a final output. XGBoost is known for its high performance in large datasets and its ability to handle imbalanced classes, which is useful for detecting rare fraudulent transactions.

## Retrieval Augmented Generation Model (RAG)

Retrieval Augmented Generation (RAG) is a sophisticated approach that integrates retrieval mechanisms with generative models to enhance the quality and relevance of generated content. By combining a retriever, which fetches pertinent information from a large corpus, with a generator that produces coherent text based on this information, RAG models can significantly improve performance on tasks requiring both detailed knowledge and creative generation (Lewis et al., 2020).

### Model Architecture

In this approach, BERT (Bidirectional Encoder Representations from Transformers) is utilized for tokenization and generating text embeddings. BERT enables effective capture of contextual information from text (Devlin et al., 2019). For managing and retrieving vector representations, FAISS (Facebook AI Similarity Search) is employed. FAISS offers efficient similarity search and clustering of dense vectors (Johnson et al., 2017). For the generative component, GPT-3.5 is used, a state-of-the-art model that leverages a transformer-based architecture to produce high-quality and coherent text based on the retrieved information (Brown et al., 2020).   
  
Model customizations and optimizations

1. Using Sentence Transformer for embedding, which provides meaningful, dense representation of text data.
2. FAISS vector store for efficient search, and capability to handle large datasets.
3. Custom scoring with high priority terms to give more importance in the similarity search for specific keywords.
4. GPT 3.5 for response generation for informative response generation.

Multiple experiments were conducted to identify which embeddings to use. I will discuss this further in next chapter along with the findings. As the first step, features were converted to embeddings and stored in the FAISS index.

A screenshot of a computer

Description automatically generated

Then FAISS index is read and used the embeddings to identify and flag anomalous transactions using Euclidean distance and cosine dissimilarity.

I implemented a similarity search method and developed a prototype generative model. When a user inputs a query, the system searches the vector store using similarity measures to determine if a transaction is likely fraudulent or genuine and provides contextual information about the transaction.

## Evaluation

All models were rigorously evaluated using several key metrics to assess their performance. Accuracy Score was calculated to measure the overall correctness of each model's predictions. Additionally, a Classification Report was generated, providing a detailed breakdown of each model's performance, including precision, recall, and F1-score.

**Precision** measures the proportion of true positive predictions among all positive predictions made by the model, reflecting its ability to avoid false positives.

**Recall** indicates the proportion of actual positives correctly identified by the model, highlighting its ability to capture true positives.

**F1-score** provides a harmonic mean of precision and recall, offering a balanced view of the model's performance, especially useful when dealing with imbalanced datasets.

Also, a confusion matrix was created to visualize the performance of model. This matrix shows the counts of true positive, true negative, false positive, and false negative predictions. This helps to understand where the model is making errors as well as where it is performing well.

To evaluate similarity search of the retriever model, below metrics were used

**Average Precision (AP)**: Average Precision is the average of the precision values calculated at each relevant item in the result list. It provides a single number that summarizes the precision of the model across all positions in the ranked list. AP is useful for summarizing how well the model performs across all relevant items, not just the top-K results. It captures both precision and recall in a single metric, reflecting the model's overall performance in ranking relevant items higher.

**Normalized Discounted Cumulative Gain (NDCG):** NDCG measures the usefulness of the results based on their position in the ranked list giving more weight to relevant items that appear earlier in the list. This metric helps in understanding how well the model orders the relevant items, which is critical for applications where the order of results impacts decision-making.

# RESULTS

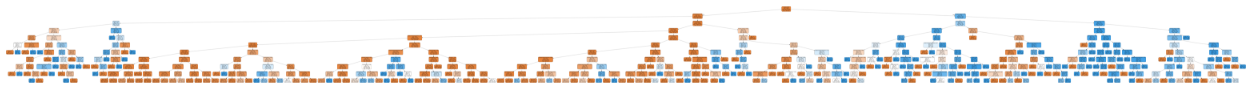
**Experiment 1:** Performing Random Forest Classifier on imbalance dataset vs SMOTE oversampled dataset and optimized model

|  |  |
| --- | --- |
|  |  |
| f | f |

**Analysis:**

Model 2 outperforms Model 1 in precision and recall for detecting fraudulent transactions, achieving slightly higher accuracy and better metrics overall. This improvement is due to Model 2’s use of SMOTE for balancing the dataset and customized configurations. Model 1, though effective, is limited by the imbalanced dataset, resulting in lower precision and recall for the minority class.

**Output:**



**Experiment 2:** Basic Neural Network model vs Customized Neural Network Model on imbalanced dataset

|  |  |
| --- | --- |
|  |  |
|  |  |

**Analysis:**

Model 2 significantly outperforms Model 1. While Model 1 achieves a high accuracy of 98.84%, it fails to detect fraudulent transactions resulting in a recall of 0% for fraud detection. In contrast, Model 2, with customized features like dropout layers and early stopping, achieves 99.50% accuracy and improved fraud detection with a precision of 0.94 and recall of 0.60. This makes Model 2 more effective at identifying both genuine and fraudulent transactions, better addressing class imbalance.A graph of different colored lines

Description automatically generated with medium confidence

**Experiment 3:** Customized NN model on imbalanced dataset vs SMOTE oversampled dataset

|  |  |
| --- | --- |
|  |  |
|  |  |

Model 1 achieves a high accuracy of 99.50%. It effectively identifies genuine transactions with a precision and recall of 1.00. But struggles with fraudulent transactions, yielding a recall of 0.60. Despite its high accuracy its performance on the minority class is limited.

In contrast Model 2 has a slightly lower accuracy of 98.59%. However, it significantly improves performance on fraudulent transactions, with a precision of 0.98 and recall of 0.99. This reflects a balanced performance across both classes. Model 2, using oversampling handles class imbalance better leading to improved detection of both genuine and fraudulent transactions.

**Experiment 4:** XGBoost on imbalanced vs SMOTE oversample dataset

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

**Analysis:**Model 1,an XGBoost classifier on an imbalanced dataset, has a high test accuracy of 99.68% but lower recall for fraud detection. Model 2, using the same XGBoost model on a SMOTE-oversampled dataset, shows slightly lower accuracies but significantly better performance for fraudulent transactions, with precision and recall both reaching 0.99 and 1.00, respectively.

**Experiment 5:** TF-IDF embedding vs BERT Embedding

|  |  |
| --- | --- |
|  |  |
|  |  |

**Analysis:** BERT embeddings offer a more comprehensive and contextually aware representation of text as represented as large vector size (768 dimensions).

**Experiment 6:** Identifying anomalous transactions by embedding cosine approach vs embedding Euclidean approach vs random forest approach on SMOTE oversampled data

|  |  |
| --- | --- |
|  |  |
|  |  |

**Analysis:**

Random Forest has high accuracy, precision, and recall, indicating robust performance.  
Embedding-Cosine and Embedding-Euclidean approaches show very low accuracy and precision but high recall, detecting many anomalies but with many false positives. The Euclidean approach is slightly more accurate than Cosine but still less precise.  
Embedding methods are overly sensitive, leading to high false positive rates and many detected anomalies compared to the Random Forest model.

|  |  |
| --- | --- |
| A chart with yellow and purple dots  Description automatically generated |  |
| cossine | eucl |

**Experiment 7:** Evaluating similarity search on FAISS index

A screenshot of a computer

Description automatically generated

**Analysis:**

**Average Precision (AP)** shows that the model is effective in identifying relevant items with high precision early in the result set.  
**NDCG** suggests that while the model performs reasonably well, but still performs low in ranking strategy, which can be improved.  
Overall, the model exhibits strong performance in terms of precision but could enhance its ranking strategy to improve NDCG and better position relevant results.

**Experiment 8:** Creating a prototype RAG model to detect transaction type (fraud or genuine)

Evaluation of generative model response is complex and not straightforward as traditional models. However, for evaluation I have parsed the exact query prompt, and the model has identified it correctly.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

However, if the query prompt is partial these results may vary.

A diagram of blue dots

Description automatically generated

Query Prompt: ‘*what is the type of A recurring transaction originated from a Teen male customer 180936571 from merchant 980657600 to category es\_sportsandtoys with considerable amount is categorized as?* ‘

Generated Response:

A black and white text

Description automatically generated with medium confidence

‘*Prediction:  
This transaction is a fraud transaction*

*Generated Response:  
Based on the provided documents, a recurring transaction originated from a Teen male customer with a considerable amount to the category es\_sportsandtoys is categorized as a fraud transaction. ‘*

References

 Association of Certified Fraud Examiners (ACFE). (2022). *2022 Report to the Nations: Global Study on Occupational Fraud and Abuse*. ACFE.

 Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.

 Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

 LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.

 Lewis, P., Perez, E., Piktus, A., Karpukhin, V., Goyal, N., Kuang, Z., ... & Riedel, S. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.

** Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.**

** Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.**

** LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521(7553), 436-444.**

**UK Finance. (n.d.). *Annual Fraud Report 2024*. [online] Available at:** [**https://www.ukfinance.org.uk/policy-and-guidance/reports-and-publications/annual-fraud-report-2024**](https://www.ukfinance.org.uk/policy-and-guidance/reports-and-publications/annual-fraud-report-2024)**.**

** Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68, 90-113.**

** Bolton, R. J., & Hand, D. J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235-255.**

** Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.**

** Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357.**

** Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.**

** Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. *IEEE Symposium Series on Computational Intelligence*.**

** Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.**

** Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.**

** Jing, H., Ngai, E. W. T., Lee, S. Y., & Xing, X. (2018). Fraud detection for online businesses: A perspective from blockchain technology. *Internet Research*, 28(5), 1068-1087.**

** Khattak, M. A., Hashmi, M., Ahmad, J., & Kang, B. H. (2022). Using Retrieval-Augmented Generation (RAG) to generate correct answers for legal case question answering. *Applied Sciences*, 12(6), 3198.**

** Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33, 9459-9474.**

** Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.**

** Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.**

** Pourhabibi, T., Ong, K. L., Saeed, A., & Yahyapour, R. (2020). Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133, 113303.**

** Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.**

** Wang, X., Zhang, Y., & Liu, W. (2021). Financial text classification using pre-trained language models. *Information*, 12(9), 364.**

** Wei, W., Hu, L., & Zhang, Y. (2019). A novel ensemble method for credit card fraud detection using optimized deep belief networks. *Soft Computing*, 23(17), 7939-7950.**

**AWS (2024) *What is Retrieval-Augmented Generation?*, AWS. Available at:** [**https://aws.amazon.com/what-is/retrieval-augmented-generation/**](https://aws.amazon.com/what-is/retrieval-augmented-generation/) **(Accessed: 24 August 2024).**

**Izacard, G. and Grave, E. (2021) ‘Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering’, *arXiv preprint arXiv:2007.01282*. Available at:** [**https://arxiv.org/abs/2007.01282**](https://arxiv.org/abs/2007.01282)**.**

**Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D. and Yih, W.-T. (2020) ‘Dense Passage Retrieval for Open-Domain Question Answering’, *arXiv preprint arXiv:2004.04906*. Available at:** [**https://arxiv.org/abs/2004.04906**](https://arxiv.org/abs/2004.04906)**.**

**Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Kulshreshtha, A., Chen, W., Bitton, J., Chaudhary, V., et al. (2020) ‘Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks’, *Advances in Neural Information Processing Systems*, 33, pp. 9459–9474.**

**Qu, Y., Xiong, C., Li, L., Liu, J., Wang, Y. and Liu, T.-Y. (2021) ‘RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering’, *arXiv preprint arXiv:2010.08191*. Available at:** [**https://arxiv.org/abs/2010.08191**](https://arxiv.org/abs/2010.08191)**.**

**Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2020) ‘Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer’, *Journal of Machine Learning Research*, 21(140), pp. 1-67. Available at: http://jmlr.org/papers/v21/20-074.html.**

**Shuster, K., Ju, D., Roller, S., Dinan, E., Boureau, Y.-L. and Weston, J. (2021) ‘Dialogue Retrieval and Generation: Better Together’, *arXiv preprint arXiv:2104.08524*. Available at:** [**https://arxiv.org/abs/2104.08524**](https://arxiv.org/abs/2104.08524)**.**

**Guyon, I. and Elisseeff, A., 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, pp.1157-1182.**

** Breiman, L., 2001. Random forests. *Machine Learning*, 45(1), pp.5-32. Available at:** [**https://link.springer.com/article/10.1023/A:1010933404324**](https://link.springer.com/article/10.1023/A:1010933404324) **[Accessed 24 Aug. 2024].**

** Liaw, A. and Wiener, M., 2002. Classification and regression by randomForest. *R news*, 2(3), pp.18-22. Available at: https://cran.r-project.org/doc/Rnews/Rnews\_2002-3.pdf [Accessed 24 Aug. 2024].**

** Zhang, Y. and Zhou, Z.H., 2013. A review on multi-class classification. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), pp.1-12. Available at:** [**https://ieeexplore.ieee.org/document/6505733**](https://ieeexplore.ieee.org/document/6505733) **[Accessed 24 Aug. 2024].**

**Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep Learning*. MIT Press.  
LeCun, Y., Bengio, Y. and Hinton, G., 2015. 'Deep learning'. *Nature*, 521(7553), pp.436-444.**

**Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). ACM.**

**Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189-1232.**

**Lewis, P., Liu, T., Goyal, N., Ruder, S., Edunov, S., Stoyanov, V., & Zettlemoyer, L. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Proceedings of the 43rd International Conference on Machine Learning (ICML 2020)*, 119, 5938-5950.**

**Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.**

**Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, 4171-4186.**

**Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. *Proceedings of the 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2327-2336.**

**Buckley, C. and Voorhees, E.M., 2000. Evaluating evaluation measures. *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pp. 25-32.**

**Jarvelin, K. and Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4), pp. 422-446.**