A black background with white text

Description automatically generated with low confidence

MSc Data Science Project

7PAM2002-0509-2023

Department of Physics, Astronomy and Mathematics

**Data Science FINAL PROJECT REPORT**

**Project Title:**

Enhancing Fraud Detection: A Comparative Study of Ensemble Learning, Machine Learning, and Retrieval-Augmented Generation

**Student Name and SRN:**

Dhanuja Tharinda Memmenda Arachchi (21062872)

Supervisor: Vito Graffagnino

Date Submitted: 29/08/2024

Word Count: Enter the word count

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science at the University of Hertfordshire.

I have read the guidance to students on academic integrity, misconduct and plagiarism information at [Assessment Offences and Academic Misconduct](https://www.herts.ac.uk/__data/assets/pdf_file/0007/237625/AS14-Apx3-Academic-Misconduct-v17.0.pdf) and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project module or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6). I have not used chatGPT, or any other generative AI tool, to write the reportor code (other than where declared or referenced).

I did not use human participants or undertake a survey in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Tharinda Arachchi

Student Name signature:

Student SRN number: 21062872

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

# Abstract

***Financial fraud is a high and increasing threat to the world economy. Fraudulent transactions are very sophisticated nowadays and present a big problem for traditional detection methods. The key purpose of this study is to investigate the possibility of Retrieval-Augmented Generation in the enrichment of complex and emerging fraud pattern detection within financial transactions and customizing existing models for better capturing patterns and increasing accuracy of predictions. Up to now, traditional models such as random forests and neural networks have been applied to fraud detection. Very often, this kind of model fails to identify new or evolving fraud strategies since it is being trained on historical data. In contrast, RAG combines retrieval-based models and generative models in a way that caters to the dynamic integration of external knowledge for context-aware analysis hence may raise the bar for the detection of novel fraud patterns. This research deals with a comparison between the performances of RAG with the traditional models in terms of accuracy, adaptability, and effectiveness. The paper also designs a prototype RAG application, which uses a vector store and generative AI for the retrieval and analysis of related transactions based on a user query. Furthermore, investigating optimal embeddings, distance metrics and configurations for model is also intended through this study. Though the RAG has flexible and dynamic retrieval capacities, precision and accuracy are matters where the traditional models perform better. However, the study demonstrates that quality embedding and the treatment of false positives make the RAG models much more effective. But such limitations are the synthesis made up of datasets and poor access to state-of-the-art embedding techniques. Future work must now concentrate on optimized retrieval mechanisms of RAG, alternative embedding methods, and the unification of RAG with advanced machine learning models to develop a hybrid approach for providing further effective fraud detection.***

## 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 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 analysing patterns within large and complex datasets. Some of the most 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 labelled 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.

# 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 enhances the detection of complex and emerging fraud patterns. However, the application of RAG in the context of financial fraud detection remains underexplored.

This study aims to address the gap by investigating the potential of RAG in financial fraud detection systems. Also, investigate the effectiveness of the traditional models by customizing and fine tuning these models to retrieve higher performance.

# 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 are underexplored in the context of non-textual datasets. Hence, there was an 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 study, 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.

Finally, I have extensively customized and fine-tuned traditional models to investigate how these techniques can enhance model performance.

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 compared 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 and creating a prototype RAG application.

**Objectives**

**Assess the Capabilities of RAG:** Investigate how Retrieval-Augmented Generation (RAG) can be utilized to identify and analyse 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 RAG application that leverages vector stores and generative AI to retrieve and analyse 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:** Conducting experiments with imbalanced dataset and oversampled dataset and see how it effects model performance

Ethical Considerations

This adheres to some important principles that guide ethics: the dataset is anonymous and genuinely ethically gathered, and both code and report are an original work. Also, it is in accordance with privacy and data security norms with respect to human life, the law, environment, and the general welfare. It underlines precision and rigour by citing all relevant articles and journals in full, which therefore demonstrates commitment to exact and reliable research. This further serves as a good example of leadership and communication through the use of ethical research techniques and presentation of results in a responsible and transparent way.

## 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 behaviour 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**

These models are designed to generate semantically meaningful sentence embeddings that can be used for tasks like similarity search by 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 and 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 data source. 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 BERT or GPT to generate coherent and contextually enriched responses (Karpukhin et al., 2020)

A diagram of a computer process

Description automatically generated

Figure 1 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 forward 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 structure dataset such like 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.

## METHODOLOGY

# Overview of Data

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

Figure 2 Overview structure of raw dataset

A screenshot of a phone

Description automatically generated

Figure 3 Datatypes of raw dataset

# 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. |

Table 1 Overview of columns

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

The data set was pre-processed to be informative following many other steps, such as using label encoding to convert categorical data into numerical representations, removing redundant or duplicate entries, handling missing or unknown values, and removing unwanted characters from the data set. Such steps are prerequisites to ensure that the dataset becomes more clean and ready for modelling.

For categorical encoding in this project, I used label encoding. Label encoding is a technique which used to convert categorical data into numerical format. Each unique category value is assigned a unique integer by allowing machine learning algorithms to process the data more effectively.  
Then, I removed ‘zipCodeOri’ and ‘zipMerchant’ columns from the dataset as it contains one unique value across the dataset. Also, I have removed the un-necessary characters in the values before parsing to modelling.

Finally, I have removed unknown values especially in customer age column which happened to be ‘U’ as value.

Next, the dataset was analysed by plotting it on a variety of attributes for an Exploratory Data Analysis (EDA).

|  |  |
| --- | --- |
| A diagram of a customer age  Description automatically generated | A graph with numbers and text  Description automatically generated with medium confidence |
|  |
| A graph with blue lines  Description automatically generated |

Figure 4 Distribution of some features of dataset

A diagram of red and blue dots

Description automatically generated

Figure 5 Distribution of fraud and non-fraud data across transaction amount

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.

A graph of blue and red bars

Description automatically generated

Figure 6 Percentage of fraud to non-fraud transactions by customer age

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

Above plot depicts that fraud transactions is common in each age group while majority recorded is in ‘0’ age group.

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 |

Figure 7 Distribution of features of fraud transactions

Below graphs and plots shows the co-relationships among each feature in the dataset corresponding to target variable.

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.

A group of graphs with numbers

Description automatically generated with medium confidence

Figure 8 Relationship of customer age, amount and fraud or not

A screenshot of a graph

Description automatically generated

Figure 9 Co-relationship heatmap

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. This is Used for text embeddings. |

Table 2 Feature engineering

One of the major challenges that this dataset has is that it is extremely imbalanced. In the context of fraud detection, this would mean a very low number of fraudulent transactions, or positive cases, in comparison with the number of legitimate ones, which are the negative cases. This can result in a set of issues related to machine learning models. Model performing more biased towards the majority class is a prominent issue.

A blue rectangular bar with red and white text

Description automatically generated

Figure 10 Imbalanced dataset

To mitigate the issue of data imbalance in the dataset, it was essential to implement such strategies that could enhance the model’s ability to accurately predict both the majority and minority class. Out of the various methods available to address this challenge, I opted to use the Synthetic Minority Over-Sampling Technique (SMOTE). SMOTE is a powerful oversampling method specifically designed to handle class imbalance. Unlike traditional methods which simply duplicates existing minority data points, SMOTE generates synthetic data points by interpolating between minority class instances.

A graph with red and blue squares

Description automatically generated

Figure 11 SMOTE Oversampled dataset

# Trained Algorithms to Analyse 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 one of the most robust ensemble learning methods. It is common in tasks such as classifications, fraud detections, and so on. During training, a number of decision trees are created, and the output is then aggregated to come up with the final decision. This process involves in two main 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 |

Table 3 Model configuration for Random Forest

### Model Customizations and Optimizations

1. Handling data imbalance problem by SMOTE oversampling and adjusting the bias towards model.
2. Custom hyper parameters usage to provide a more balance between computational efficiency and performance (i.e: n\_estimators, max\_depth, min\_samples\_split and min\_samples\_leaf)
3. Label encoding done to utilize categorical features in model predictions.
4. Standard scaling to standardize features to have zero mean and unit variance.
5. Feature importance calculation.

# Neural Network

Basically, NN models are inspired by the functioning of the human brain: recognizing patterns and making a decision based on input data. These models have multiple layers connected with neurons, and every connection is associated with a weight which is adjusted while training. Passing the data through the created layers makes the network learn. That is how it models complicated relationships and performs classification and regression tasks efficiently.

### 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 optimized distributed gradient boosting library structured to be highly efficient in terms of running speed and model execution performance (Chen & Guestrin, 2016). Key novelties of XGBoost over other gradient boosting techniques include advanced computational techniques, such as regularization, parallel computing, and optimization.

### 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 contents. By integrating a retriever, it fetches relevant information from a large corpus. With a generator which is capable in producing coherent and context rich response based on the retrieved results, RAG model can significantly improve the performance of tasks which requires both detailed knowledge and creative generation (Lewis et al., 2020).

### Model Architecture

In this study, 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).   
Also, a different embeddings method like OPEN AI, which is external API equipped to compare the effectiveness of text embeddings.

Model customizations and optimizations

1. Usage of various embedding methods like BERT, OPEN AI, TF-IDF and identified optimal for the model, 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.

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

Also, I implemented a similarity search method for retrieval and developed a prototype generative model to showcase how RAG can be applied to fraud detection. This involves necessary pre-processing, text embeddings, vectorization, similarity search with input query prompt and context rich response generation to identify whether the transaction is fraudulent or not.

A screenshot of a computer

Description automatically generated

Figure 12 Text Embeddings

As per the above image, additional text column is describing the behaviour of each transaction. Then these texts were converted to embeddings using BERT tokenizers. Next, these were added to FAISS vector store/index.

When a user inputs a query, the system searches the vector store using similarity measures finding and ranking based on the distance metric and use the results to determine if a transaction is likely fraudulent or not. Then provides contextual information about the transaction with the usage of generative models.  
Similarity search is an approach to find the most nearby neighbour in simple terms. This typically means that finding the vector in a vector space that are close to the query vector based on any distance metric. Widely used distance metrics are Euclidean and Cosine similarity. In this approach, I have utilized both these metrics.

# Evaluation

All the models were tested based on a couple of key metrics that gives an overview of how well each model is performing. Then, calculate the Accuracy Score, which measures what proportion of predictions a model gets correct. Also, generate a Classification Report, which gives detailed reports on how every model performs in terms of precision, recall, and F1-score.

**Precision** measures the fraction of the true positive prediction in all positive predictions predicted by the model, which indicates how well it is at avoiding false positives.

Equation 1 Precision

**Recall** is the proportion of actual positives that the model has correctly classified.

Equation 2 Recall

**F1-score** gives both the harmonic mean of precision and recall, therefore providing a well-balanced view of the model performance. It is useful, much like accuracy, especially on imbalanced datasets.

Equation 3 F1 Score

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)** The average precision is defined as the average of precisions at every relevant position in the ranked list. It produces a single number summarizing the precision of the model across all ranked positions. AP gives a summary of how well the model does across all relevant items and not just the top-K results. It combines precision and recall in one number, thus indicating the ranking performance of the model.

**Normalized Discounted Cumulative Gain (NDCG):** NDCG measures the usefulness of the results based on the position in the ranked list by 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 AND ANALYSIS

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

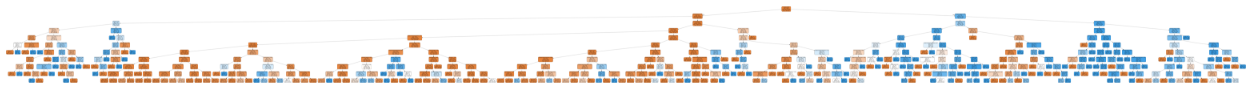
|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated  Figure 13Basic model with Imbalanced dataset | A screenshot of a computer  Description automatically generated  Figure 14 Customized model, SMOTE dataset |

### Analysis:

Model 2 performs better than model 1 in precision and recall toward fraud detection, with slightly greater accuracy and generally better metrics. This is because Model 2 was fitted with SMOTE for dataset balancing and customized configurations. Model 1, although effective, is constrained with this imbalanced dataset and hence gives lower precision and recall values for the minority class.

### Output:

Below is the generated classification tree. Original Tree can be found in GitHub repository under reports folder.



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

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated  Figure 15 Basic NN on imbalanced dataset | A screenshot of a computer  Description automatically generated  Figure 16 Customized NN on imbalanced dataset |

### Analysis

Model 2 significantly outperforms Model 1. Model 1 achieves a very high accuracy of 98.84%, but it misses fraud transactions, and hence the recall for fraud detection is 0%. In contrast, Model 2 uses customized features like dropout layers and early stopping to get an accuracy of 99.50% and significantly improved fraudulent transaction detection with a precision of 0.94 and recall of 0.60. This makes Model 2 more effective at classifying real and fraudulent transactions and better handling class imbalance.

A graph of different colored lines

Description automatically generated with medium confidence

Figure 17 Loss and Accuracy plot

As per the training and validation loss, model is learning effectively and generalizing well to the validation data. The model’s performance seems stable and improving without overfitting.

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

|  |  |
| --- | --- |
| Figure 18 Customized NN model on imbalanced dataset | Figure 19 Customized NN model on SMOTE dataset |

### Analysis

The first model attains a high level of accuracy, 99.50%. It can identify genuine transactions with both precision and recall of 1.0. It really suffers on fraudulent transactions but leads to a recall of 0.60 for such samples. In this way, it performs only well on the minority class, though having a high accuracy.

Just a little lower on the accuracy, Model 2 gives an impression of 98.59% with wholly significant improvement on the fraudulent transaction that belonged to a high precision of 0.98 and 0.99 in recall; it represents a balanced performance across the classes. Model 2 using oversampling thus handles the class imbalance better to detect both genuine and fraudulent transactions better.

Experiment 4: XGBoost on imbalanced vs SMOTE oversample dataset

|  |  |
| --- | --- |
| A screenshot of a computer screen  Description automatically generated | A screenshot of a computer  Description automatically generated |
| A graph of confusion matrix  Description automatically generated  Figure 20 XGBoost on imbalanced data | A blue squares with white text  Description automatically generated  Figure 21 XGBoost on SMOTE data |

Analysis:Model 1: an XGBoost classifier over an imbalanced dataset with a very high test accuracy of 99.68%, although this has lower recall for fraud detection. Model 2, with the same XGBoost model but on a SMOTE-oversampled dataset, gives somewhat lower accuracies but very good performance for fraudulent transactions, where precision and recall reach 0.99 and 1.00, respectively.

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

|  |  |
| --- | --- |
| Figure 22 Shape of TF-IDF | Figure 23 Shape of BERT |

**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 using distance metrics.

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

Figure 24 Embedding cosine vs Euclidean distance

### Analysis:

Majority of the performance metrics in terms of accuracy, precision, recall, and F1-score are higher for the Embedding-Cosine Approach. Moreover, it also detects more anomalies. That means it is, in general, more effective and robust for the task of flagging anomalous transactions based on distance measures.

Embedding-Euclidean Approach: This showed slightly worse performance metrics compared to the Cosine Approach. Still, it could prove useful depending on the application requirements—where, in particular, a lower rate for anomaly detection is desired.

In most cases, the Cosine Approach should hence be the better choice due to its increased accuracy and, more importantly, well-balanced performance. However, the final choice between these approaches could also depend on specific needs and sensitivity trade-offs of anomaly detection.

|  |  |
| --- | --- |
| A chart with yellow and purple dots  Description automatically generated  Figure 25 Cosine approach | A graph with yellow dots  Description automatically generated  Figure 26 Euclidean approach |

Experiment 7: Identifying anomalous transactions by embedding cosine approach vs embedding Euclidean approach using BERT vs OPEN AI embeddings

|  |  |
| --- | --- |
| Figure 27 BERT Embeddings | Figure 28 OPEN AI Embeddings |

In this experiment I have utilized BERT and OPEN AI embedding methods to generate embeddings and investigated efficiency of identifying anomalies using distance matrix.

### Analysis

It is evident that Model 1 which uses BERT embeddings, and by all parameters, such as accuracy, precision, recall, and F1-score, the Embedding-Cosine Approach goes higher than the Embedding-Euclidean Approach. Moreover, it is able to detect more unique anomalies: 228 against 12 by the second approach. Contrary to this, Model 2 with OpenAI embeddings shows the Embedding-Euclidean Approach performing better than the Embedding-Cosine Approach in terms of accuracy with a value of 0.576 against 0.45 and in terms of precision with a value of 0.578 against 0.468, with an F1-score a little higher. The Embedding-Cosine approach with OpenAI embeddings, however, detects more of the unique anomalies: 204 against 25. Conclusively, BERT embeddings generally give better performance metrics, while OpenAI embeddings in the Euclidean approach give better accuracy and precision. The choice about which model to use may be, therefore, dependent on whether the priority is higher detection sensitivity or better overall performance on the classification.

Experiment 8: Evaluating similarity search on FAISS index

A screenshot of a computer

Description automatically generated

Figure 29 Evaluation of similarity search

### 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 9: 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

Figure 30 Evaluation of RAG model (full text)

A screenshot of a computer

Description automatically generated

Figure 31 RAG Prediction (full text)

However, if the query prompt is partial these results may vary. A diagram of blue dots

Description automatically generated

Figure 32 2D representation of embeddings

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

Figure 33 Generative response from RAG model

‘*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. ‘*

# Conclusion

This research explored the capabilities of Retrieval Augmented Generation (RAG) in identifying fraudulent transactions and compared its effectiveness with traditional models. Furthermore, these traditional models were customized and fine-tuned to increase accuracy and predictions. Also, experiments were carried out to see how class imbalance effect model performance.

While RAG shows adaptability and dynamic retrieval of relevant information, traditional models like Random Forest and XGBoost shows higher accuracy, precision and recall in fraud detection. The developed prototype model using vector store for efficient retrieval and generative AI for generative context rich response demonstrates the practical application of RAG. However, the effectiveness of RAG model depends on the quality of embeddings and handling of false positives.

# Limitations

This study has several limitations. The dataset used in this study was synthetically generated. Hence, it lacks the complexities and nuances of real-world transactions. Additionally, there was a limited access to more advanced embedding methods like ‘text-embedding-ada-003’. These methods can generate higher dimensional and higher quality embeddings which potentially increase the performance of RAG application. Furthermore, training a RAG application model is computationally expensive. This was a major challenge in optimizing and scaling the model effectively.

# Future Work

Future research should aim to optimize RAG’s retrieval mechanism to enhance precision and reduce false positives. Exploring alternative embedding methods and ranking strategies is also required. Integrating RAG with advanced machine learning models and developing a hybrid approach would also be worth of investigating.