## Final Report Sections

---

## 1. Background

- **What to Include:**
  - **General Problem:** Define fraud detection as a critical challenge in finance and explain how bias impacts model outcomes.
  - **Challenges:** Discuss issues like class imbalance, bias in datasets, and the ethical implications of unfair AI models.
  - **Literature Survey:** Mention key studies (e.g., Kamalaruban et al., Pombal et al.) that address fairness metrics and bias mitigation techniques.
  - **Significance:** Explain why addressing fairness in fraud detection is vital for ethical AI and societal trust.

Currently, an issue faced by banks is fraudulent bank account applications, creating the need for fraud detection systems. Two key pieces of literature contributed to the motivation of this study: Kamalaruban et al.'s *"Evaluating Fairness in Transaction Fraud Models: Fairness Metrics, Bias Audits, and Challenges"* and Pombal et al.'s *"Understanding Unfairness in Fraud Detection Through Model and Data Bias Interaction."* In machine learning models, it is important to ensure fairness, or the lack of discrimination based on sex, race, and religion. However, training datasets are often biased towards certain demographic groups of people. Bias in fraud detection systems stems from class imbalances in training data and shortcomings in algorithmic design techniques. Since machine learning algorithms blindly embrace these biases while making their decisions, biased models harbor the capacity to worsen social inequities (Pombal et al. 1). In this study, potential consequences of bias include the unjust denial of bank account access to those of certain demographic groups, leaving banks susceptible to financial and legal repercussions. (Kamalaruban et al. 1). By addressing fairness in fraud detection systems in this study, we hope to alleviate these consequences.

---

## 2. Introduction

- **What to Include:**
  - **Project Goals:** State the purpose of the project, such as creating fair and accurate fraud detection models.
  - **Datasets:** Briefly introduce the Bank Account Fraud suite and its privacy-preserving features.
  - **Methodologies:** Summarize your approaches, including EDA, data preprocessing, and bias mitigation techniques.

- ○ **Fairness Considerations:** Highlight the use of fairness metrics like Demographic Parity Difference and Equalized Odds Difference.

Our goal is to develop a hybrid fraud detection model that maximizes selected classification metrics while minimizing bias by incorporating bias mitigation techniques in the pre-processing, in-processing, and post-processing stages.

The dataset employed in this study is the Bank Account Fraud Suite (BAF). While the dataset contains valuable data for training and testing, issues posed by the BAF include class imbalances, bias, and maintaining applicant privacy. To protect sensitive information, attributes that can yield the identity of applicants, specifically age and income, are omitted. Furthermore, the BAF suite is perturbed using Laplacian noise as an additional method of privacy insurance.

In this study, a pre-processing method to reduce bias is reweighting, where a weight is given to each observation by calculating the ratio of its population to sample proportion (Qian et. al.) Furthermore, synthetic minority oversampling for datasets with numerical and categorical features (SMOTENC) also helps mitigate bias in this stage, which creates artificial samples for minority groups while removing observations belonging to the majority class. In the processing phase, bias is reduced through adversarial debiasing learning, which runs two opposing models: a main and an adversarial model. The main model tries to predict the classification label Y using input X while the adversarial model seeks bias patterns in the main model. In the post-processing stage, a threshold optimizer with an equalized odds constraint finds the best threshold quantity to optimize the equalized odds fairness metric. Lastly, the equalized odds and demographic parity fairness metrics ultimately quantify the bias present in the model.

---

## 3. Methodologies

### 3.1 Exploratory Data Analysis (EDA)

- **What to Include:**
  - ○ Describe the initial analysis of the dataset to identify trends, distributions, and anomalies.
  - ○ Discuss tools used (e.g., histograms, box plots, correlation matrices) to understand data characteristics.

We began our analysis with six datasets from the Bank Account Fraud (BAF) suite, the Base dataset and Variants I through V. Each dataset consists of  32 features comprising both numerical and categorical variables. Our primary focus was on the target variable fraud_bool, which labels applications as fraudulent (1) or legitimate (0). A significant class imbalance was immediately evident, with fraudulent applications constituting less than 2% of the total records, as showcased in Figure 1. To further analyze the data, we used box plots and histograms to observe the distributions of numerical features, which helped us spot the outliers and understand the overall data spread, which were important factors for our subsequent modeling steps. Correlation matrices revealed strong relationships between name_email_similarity and fraud likelihood, and between device_distinct_emails_8w and legitimate applications. These key insights guided our feature selection and engineering, ensuring we prioritized the most predictive variables. To interpret the data effectively, we utilized Matplotlib and Seaborn to create informative charts, while Pandas and NumPy were essential for efficient data handling and statistical analysis. Additionally, correlation analysis played a key role in identifying multicollinearity among features, informing our strategy for building robust predictive models.

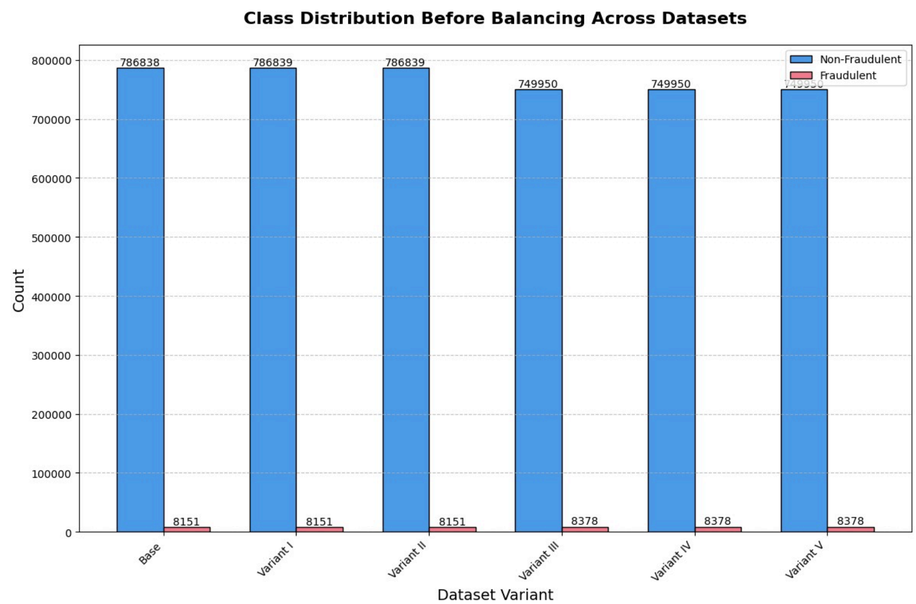



Figure 1: Bar graph depicting class imbalance in the datasets

### 3.2 Data Processing

- **What to Include:**
  - Explain steps for handling missing values, outliers, and redundant data.
  - Describe feature scaling, categorical variable encoding, and other transformations performed.

We began by verifying that each dataset included the essential columns: fraud_bool, month, and employment_status and identified missing values represented as -1 in various features. To simulate real-world scenarios where models predict future events based on historical data, we shuffled the datasets and performed a temporal split into training and testing sets using the month feature. To address missing values, we employed various imputation methods: mean imputation for numerical features, replacing missing records with the mean of the respective feature to maintain the overall distribution without significant bias; mode imputation for categorical features, substituting missing values with the most frequently occurring category to preserve the mode of the existing data; and, for certain features where relationships with other variables could provide better estimates, regression-based imputation leveraging the predictive power of correlated features. Facing a significant class imbalance, with fraudulent applications vastly outnumbered by non-fraudulent ones, we initially downsampled the majority class to match the minority class size, ensuring equal representation in the training data. To further refine the dataset, we applied the Synthetic Minority Oversampling Technique for Nominal and Continuous Features (SMOTENC), generating synthetic samples for the minority class while preserving relationships between numerical and categorical features, effectively balancing the class distribution. For feature encoding and scaling, using an Ordinal Encoder, we transformed categorical variables into integer format, making the data compatible with machine learning algorithms that require numerical input, and standardized all numerical features using Z-score normalization via StandardScaler, adjusting them to have a mean of zero and a standard deviation of one to aid in the efficient convergence of the algorithms. Lastly, to promote fairness and mitigate potential bias associated with the sensitive attribute employment_status, we implemented a reweighting technique using the AIF360 library, which adjusted the sample weights in the training data to counteract group size disparities, ensuring that the model did not favor any particular group during training.

### 3.3 Model Training

- **What to Include:**
  - List and justify the models chosen (e.g., Logistic Regression, KNN, Random Forest).
  - Explain why each model is suitable for the task of fraud detection and its ability to handle imbalanced data.

We used nine different machine learning models in our project to detect fraud while ensuring fairness across different demographic groups. Logistic Regression was employed for its ability to classify transactions as fraudulent or legitimate, giving us clear, probability-based predictions. It also shows how features like transaction amount and user behavior influence the likelihood of fraud. K-Nearest Neighbors (KNN) worked by comparing new transactions to the most similar ones in the dataset, helping to catch patterns based on proximity, which is useful for identifying fraudulent transactions similar to known fraud cases. Naive Bayes was fast and efficient at classifying transactions by calculating the probability of fraud based on feature combinations, making it ideal for processing large datasets quickly. Random Forest combined the results of multiple decision trees to make stronger and more reliable predictions, handling complex relationships between features and reducing overfitting. Support Vector Machine (SVM) was effective in finding the optimal decision boundary between fraudulent and legitimate transactions, especially when the patterns were non-linear. The SVM also incorporated fairness constraints to prevent bias toward any particular demographic group. Neural Networks, with their ability to learn from intricate patterns in the data, helped in identifying subtle fraud signals, while fairness-aware techniques ensured that the network didn't learn bias from the data.

Additionally, we used Gradient Boosting Machines (GBM) and XGBoost, both of which combined weak learners (decision trees) to form strong fraud detection models. GBM iteratively corrected errors made by previous trees, improving accuracy over time. XGBoost, being optimized for speed and performance, was particularly useful for handling large datasets efficiently. Finally, LightGBM, a faster alternative to XGBoost, was used for its ability to handle large-scale data while maintaining good performance and speed. Like the other models, LightGBM was trained with fairness considerations to ensure that the predictions did not unfairly favor certain demographic groups, thus ensuring that the fraud detection system worked equally well for all users. All models were carefully trained to balance both accuracy and fairness, ensuring the system was both reliable and equitable.

### 3.4 Bias and Fairness

- **What to Include:**
  - Define fairness in machine learning and why it matters.
  - Introduce fairness metrics used in the project and explain their relevance.

Bias in Machine Learning and AI refers to systematic and unfair differences in the outcomes of models between diverse demographic groups.  Fairness, on the other hand, is the practice of ensuring that these outcomes are equitable for all groups, irrespective of attributes such as gender, race, or employment status. Mitigating bias is an essential step, especially in domains like fraud detection, as biased models can preserve existing inequalities, leading to unfair treatment of certain groups.

Addressing bias improves the ethical use of AI while also enhancing the model reliability and societal trust. The primary objective of this study is to evaluate the performance and fairness of various machine learning models across diverse datasets, with a particular emphasis on fairness metrics. Fairness metrics help us make the model outcomes equitable across various demographic groups, which is crucial in building trustworthy AI systems, particularly in sensitive areas like fraud detection.

---

## 4. Experiment Setup

### 4.1 Dataset

- **What to Include:**
  - Describe dataset attributes (e.g., number of features, sensitive attributes like `employment_status`).
  - Discuss dataset variants (Base, I–V) and their purpose in evaluating fairness and robustness.

Our datasets comprised 32 features pertinent to bank account applications, divided into numerical and categorical variables. Numerical features such as income, transaction_amount, and account_age offered quantitative insights into applicants' financial profiles and histories. Categorical features like employment_status, device_type, and application_channel provided qualitative data that could influence the likelihood of fraud. The sensitive attribute employment_status by identifying it as a potential source of bias and handling it during preprocessing and model evaluation to ensure fairness in our analysis. The target variable was fraud_bool, indicating whether an application was fraudulent (1) or legitimate (0).

To thoroughly assess our model's performance and fairness under varying conditions, we utilized several dataset variants from the BAF suite. The Base Variant served as our benchmark. Variant I introduced a higher group size disparity, affecting the balance between demographic groups and challenging the model's ability to generalize. Variant II contained higher prevalence disparity with varying fraud rates across groups, testing the model's robustness to uneven class distributions. Variant III offered better separability for one group, potentially enhancing detection capabilities for that group but risking bias against others. Variant IV featured higher prevalence disparity in the training data, which could impact the model's generalization to unseen data. Lastly, Variant V provided better separability in training for one group, possibly leading to overfitting and reduced performance on the test set.

### 4.2 Exploratory Analysis and Data Pre-processing

- **What to Include:**
    - Summarize findings from EDA and preprocessing, focusing on how the dataset was prepared for training.
    - Include steps like scaling, encoding, and addressing class imbalance.

During our exploratory data analysis (EDA), we uncovered key insights that shaped our data preparation and modeling approach. We observed strong correlations between features such as income and account_age, which guided our feature engineering efforts and helped us select the most predictive variables. The significant class imbalance between fraudulent and legitimate applications highlighted the need for balancing methods. This finding led us to implement methods like downsampling and SMOTENC to ensure the model learned effectively from both classes.

In preparing the datasets for model training, we addressed missing values by applying mean imputation for numerical features and mode imputation for categorical features, ensuring data completeness without introducing significant bias. To achieve a balanced training dataset, we combined downsampling of the majority class with SMOTENC for the minority class, improving the model's ability to detect fraudulent applications. We standardized numerical features using Z-score normalization (StandardScaler) and transformed categorical variables into integer format using ordinal encoding, making the data suitable for machine learning algorithms. To promote fairness and mitigate potential bias associated with the sensitive attribute employment_status, we implemented re-weighing techniques that adjusted sample weights in the training data. The final processed datasets were free of missing values and balanced in class distribution. While we retained employment_status for fairness evaluation, we excluded it from the feature set used in model training to prevent any inadvertent bias.

### 4.3 Models

- **What to Include:**
  - Provide details on the models evaluated, their configurations, and training techniques.
  - Justify model selection and highlight specific strengths for fraud detection.

We evaluated nine models for fraud detection: Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Random Forest, Support Vector Machine (SVM), Neural Networks, Gradient Boosting Machines (GBM), XGBoost, and LightGBM. These models were trained on pre-processed transaction data, with hyperparameters optimized using techniques like grid search and random search for improved performance. Logistic Regression was chosen for its simplicity and interpretability, providing clear insights into feature importance. KNN was used for detecting patterns based on transaction similarity, while Naive Bayes provided efficient classification for large datasets. Random Forest improved prediction accuracy by combining multiple decision trees and was optimized by adjusting the number of trees, tree depth, and feature selection criteria. SVM was applied to separate non-linear data, using the RBF kernel, with hyperparameters such as the regularization parameter (C) and kernel coefficient (gamma) tuned for better generalization. Neural Networks utilized a multi-layer perceptron architecture with ReLU activation functions to capture intricate patterns, and were optimized through techniques like early stopping to prevent overfitting. GBM and XGBoost were selected for their high predictive power, where learning rates, tree depths, and subsampling ratios were tuned for better accuracy, while LightGBM provided fast and scalable solutions by using histogram-based algorithms, with tuning focused on the number of leaves and learning rates. All models were evaluated based on accuracy, precision, recall, ROC-AUC, and fairness, ensuring effective fraud detection with minimal bias.

---

## 5. Bias

### 5.1 Bias Concerning Fraud Detection

- **What to Include:**
  - Discuss the origins and implications of bias in fraud detection systems.
  - Explain the role of sensitive attributes (e.g., `employment_status`) in creating unfair outcomes.

### 5.2 Bias Metrics

- **What to Include:**
  - Describe fairness metrics like Demographic Parity Difference and Equalized Odds Difference.
  - Explain how these metrics were calculated and their significance in evaluating fairness.

### 5.3 Bias Mitigation Techniques

To effectively address and reduce the bias in our machine learning models, we have employed a combination of pre-processing, in-processing, and post-processing bias mitigation techniques. Each approach targets different stages of the machine learning modeling pipeline, providing a comprehensive strategy to enhance fairness while also maintaining or improving predictive performance.

#### 5.3.1 Pre-processing Techniques

Pre-processing techniques involve modifying the training data to eliminate or reduce bias before model training. We have utilized **Reweighing**, which adjusts the weights of training samples based on their group membership and class labels. This ensures that the underrepresented or disadvantaged groups have a proportionate influence on the model training, promoting balanced representation.

Additionally, we have applied **SMOTENC (Synthetic Minority Over-sampling Technique for Nominal and Continuous)** to generate synthetic samples for minority classes while preserving the integrity of categorical features, including sensitive attributes such as employment_status. This approach not only addresses the class imbalance but also maintains the discrete nature of categorical variables, preventing the introduction of synthetic biases.

#### 5.3.2 In-processing Techniques

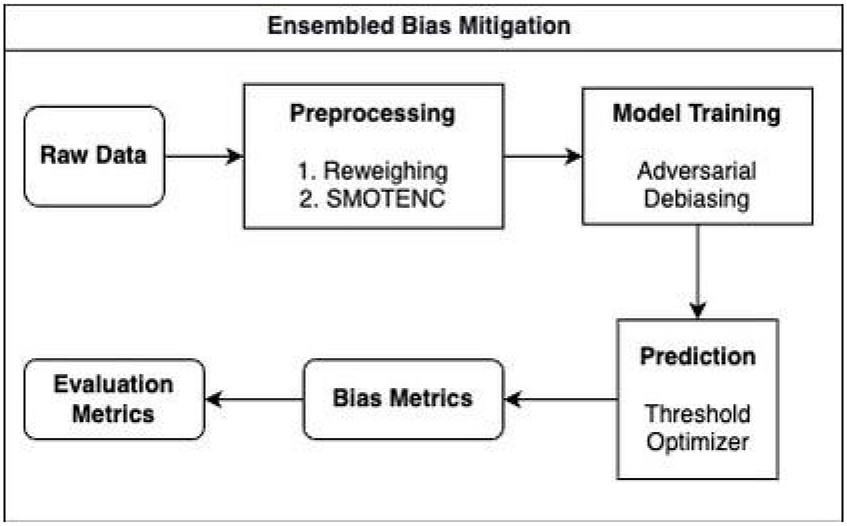
In-processing techniques integrate fairness constraints directly into the model training process. We have implemented **Adversarial debasing** using the AIF360 library, specifically tailored for the Neural Network (MLPClassifier) model. This method incorporates an adversarial network that penalizes the model for biased predictions, encouraging the primary model to produce fairer representations. Adversarial debasing effectively reduces bias without significantly compromising accuracy by optimizing for both predictive performance and fairness.

#### 5.3.3 Post-processing Techniques

After training, post-processing techniques adjust the model's predictions to meet the desired fairness criteria. We have employed the **Threshold Optimizer** from the Fairlearn library to calibrate decision thresholds based on sensitive attributes like employment_status. This technique ensures that the model's predictions adhere to fairness constraints such as equalized odds or demographic parity by adjusting the point at which predictions switch from negative to positive. While this method can improve fairness metrics, it may also lead to slight reductions in overall predictive performance as the decision boundaries are altered to achieve equitable outcomes.

### 5.4 Ensembled Bias Mitigation Architecture

To further enhance the effectiveness of bias mitigation, we developed an **Ensembled Bias Mitigation Architecture** that synergistically combines pre-processing, in-processing, and post-processing techniques. This integrated approach leverages the strengths of each mitigation stage to create a more robust framework for addressing bias throughout the modeling pipeline.




**Architecture Overview:**

1. **Pre-processing Layer - Reweighing and SMOTENC:** The initial layer involves applying Reweighing to adjust sample weights and SMOTENC to balance class distributions while maintaining categorical feature integrity. This layer ensures that the training data is fair and representative before any model training begins.
2. **In-processing Layer - Adversarial Debiasing:** Following pre-processing, Adversarial Debiasing is employed within the model training process. By integrating an adversarial network, this layer actively works to minimize bias by discouraging the model from learning biased representations related to the sensitive attribute.
3. **Post-processing Layer - Threshold Optimizer:** After the model has been trained, the Threshold Optimizer fine-tunes the decision thresholds to align the model's predictions with the desired fairness criteria. This final layer ensures that the output predictions meet fairness standards without necessitating further model adjustments.

The ensembled bias mitigation architecture offers several key benefits. Addressing bias at multiple stages of the modeling process such as pre-processing, in-processing, and post-processing ensures comprehensive bias reduction, minimizing discrepancies, and leads to more fair outcomes. This integrated approach leverages the strengths of each mitigation technique to enhance fairness without compromising predictive performance, maintaining high accuracy while significantly reducing bias.

Additionally, the architecture demonstrates robustness across various models and datasets, ensuring uniform fairness improvements regardless of underlying data distributions or model complexities. Together, these advantages make the ensembled approach a powerful framework for developing fair and reliable machine-learning models.

Overall, the Ensembled Bias Mitigation Architecture with the strategic implementation of pre-processing, in-processing, and post-processing bias mitigation techniques provided a robust framework for enhancing the fairness of our machine learning models. These methods collectively addressed the bias at various stages of the modeling pipeline, ensuring more equitable and trustworthy outcomes in fraud detection applications.

---

## 6. Results

- **What to Include:**
    - **Performance Metrics:** Present results for accuracy, F1 score, and ROC-AUC for each model and dataset variant.
    - **Fairness Metrics:** Include DPD and EOD values, comparing fairness across models and datasets.
    - **Visualization:** Use plots (e.g., bar charts, confusion matrices) to enhance clarity and comparison.

**6. Results**

The evaluation of our fraud detection models over various dataset variants revealed significant insights of both their performance and fairness. High-performing models like Random Forest, LightGBM, and XGBoost consistently achieved exceptional Accuracy, F1 Scores, and ROC-AUC Scores across multiple datasets, demonstrating strong classification capabilities. However, these models also exhibited elevated Demographic Parity Difference (DPD) and Equalized Odds Difference (EOD) values, particularly in datasets with higher group size or prevalence disparities, indicating substantial bias in their predictions.

Conversely, the K-Nearest Neighbors (KNN) along with Random Forest models also stood out for their exceptional fairness metrics, consistently showing the lowest DPD and EOD values across most dataset variants. While both maintained moderate Accuracy and F1 Scores, their ability to produce more equitable outcomes made them a favorable choice in scenarios where fairness is paramount. Logistic Regression, SVM, Gaussian Naive Bayes, and Bernoulli Naive Bayes models demonstrated a balance between performance and fairness, though they generally exhibited higher bias metrics compared to KNN and Random Forest.

The implementation of the ensembled bias mitigation architecture, which integrates pre-processing, in-processing, and post-processing techniques, effectively reduced bias across different models and datasets. Pre-processing methods like Reweighing and SMOTENC successfully balanced class distributions and adjusted sample weights, leading to lower bias metrics in models such as Logistic Regression and Random Forest without significant drops in performance. In-processing with Adversarial debiasing especially reduced bias in Neural Network models, achieving fairer predictions while maintaining competitive accuracy. Post-processing using the Threshold Optimizer further aligned fairness metrics for models like SVM and LightGBM, ensuring equitable outcomes with minimal impact on overall performance.

Overall, the ensembled approach provided a comprehensive framework that enhanced fairness across the modeling pipeline without compromising the predictive integrity of the models.

---

## 7. Conclusion

- **What to Include:**
  - Summarize key findings, including trade-offs between accuracy and fairness.
  - Highlight which models performed best for different priorities (e.g., KNN for fairness, Random Forest for accuracy).
  - Discuss how your results align with or diverge from prior literature.
- **Key Findings:**
  - **Effectiveness of Preprocessing Techniques:** The combination of downsampling, SMOTENC, and Reweighing significantly improved both the detection of fraudulent applications and the fairness of the models.
  - **Model Performance:**

### 7. Conclusion

In conclusion, we accomplished our proposed solution and objectives by developing a fraud detection system with an ensembled bias mitigation architecture enclosing pre-processing, training, and evaluation stages. Bias is inevitable in real-life intricate scenarios, this integrated approach effectively reduces bias related to sensitive attributes such as employment_status while also maintaining comparable classification metrics across various models. By strategically applying Reweighing and SMOTENC during pre-processing, incorporating Adversarial Debiasing in the training phase, and utilizing the Threshold Optimizer during post-processing, our system ensures both fairness and high predictive performance. The ensembled bias mitigation architecture proved to be a robust framework, promoting unbiased and trustworthy outcomes in our fraud detection applications.

---

## 8. Future Work

- **What to Include:**
  - Suggest advanced techniques for bias mitigation, such as adversarial debiasing or ensemble methods.

- Propose additional evaluations for underexplored sensitive attributes or other datasets.
- Mention ideas for enhancing real-world applicability, like stakeholder engagement and ongoing monitoring.

### 8. Future Work

Moving forward, we plan to integrate more advanced bias mitigation techniques to make our fraud detection models even fairer and more robust. This includes doing more market research and exploring the use of Large Language Models (LLMs) and Generative Adversarial Networks (GANs) to develop smarter ways of handling bias. By leveraging LLMs, we can create detailed methods such as Teacher-Student models like ChatGPT's O1-Preview to identify and address subtle biases present in both textual and structured data.

Additionally, we aim to implement ensemble methods that combine various bias mitigation strategies, such as adversarial debiasing and fairness-aware algorithms, to effectively reduce disparities across different demographic groups. Expanding our toolkit to include meta-learning and reinforcement learning-based fairness techniques will also allow our models to adapt dynamically to changing bias patterns, ensuring they remain natural over time.

Moreover, we intend to conduct further evaluations focusing on underexplored sensitive attributes and a wider variety of datasets to ensure our models are generalizable and effective in diverse environments. By examining factors like age, geographic location, and socioeconomic status, we can gain a more comprehensive understanding of how our models perform across different societal segments. To enhance real-world applicability, we plan to engage with stakeholders, including industry experts and affected communities, to gather valuable insights and ensure our models align with ethical standards and societal expectations. Implementing continuous monitoring systems and retraining mechanisms will be essential for maintaining fairness as new data and challenges emerge. Through these initiatives, we aim to upgrade our fraud detection systems that are not only accurate and reliable but also fair and trustworthy in a variety of dynamic environments.

---

## 9. References

- **What to Include:**
  - Cite all datasets, tools, and relevant studies, following a standard citation style (e.g., IEEE, APA).
  - Include links to the BAF dataset and any GitHub repositories used.

1. Blow, Christina Hastings, et al. “Comprehensive Validation on Reweighting Samples for

Bias Mitigation via AIF360." *Arxiv*, 19 Dec. 2023, arxiv.org/html/2312.12560v1.

2. "Threshold Optimization - (Machine Learning Engineering) - Vocab, Definition, Explanations." *Fiveable*, library.fiveable.me/key-terms/machine-learning-engineering/threshold-optimization. Accessed 1 Dec. 2024.
3. Fernandez, Franklin Cardenoso. *Bias Mitigation Strategies and Techniques for Classification Tasks*, Holistic AI, 8 June 2023, www.holisticai.com/blog/bias-mitigation-strategies-techniques-for-classification-tasks.
4. "How to Mitigate Bias in Machine Learning Models." *Encord*, 8 Aug. 2023, encord.com/blog/reducing-bias-machine-learning/.
5. Jesus, Segio, et al. "Turning the Tables: Biased, Imbalanced, Dynamic Tabular Datasets for ML Evaluation." Arxiv, 14 Nov. 2022, https://arxiv.org/abs/2211.13358. Accessed 14 Sept. 2024.
6. Kamalaruban, Parameswaran, et al. "Evaluating Fairness in Transaction Fraud Models: Fairness Metrics, Bias Audits, and Challenges." *arXiv.Org*, 6 Sept. 2024, arxiv.org/abs/2409.04373.
7. Pombal, José, et al. "Understanding Unfairness in Fraud Detection through Model and Data Bias Interactions." *arXiv.Org*, 13 July 2022, arxiv.org/abs/2207.06273.

BAF Suite Link: https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

GitHub Link: https://github.ncsu.edu/dpendya/engr-ALDA-Fall2024-P22