Domain Background:

Uganda has adopted the use of internet banking, using smart phones, agent banking and through portals that have integrated APIs to transact directly with their customers to streamline access to financial services. In fact, most of the Uganda's population, both rural and urban population have quickly adopted internet banking as opposed to traditional banking,with over 11 million users on only MTN network from Uganda (Ali et al., 2020). They use it for savings depositing, microfinance financial management, retail sales, and business-to-business transactions. More specifically, services such as sending and receiving money, paying for utilities, savings schemes, money transfers, and pension payments use internet banking widely in Uganda (Ali et al., 2020). Despite its benefits, there has been a seriesof concerns in relation to internet banking in Uganda. These include security concerns from identification and validation of account holders, poor network coverage, high transaction costsas reported by certain parts of the population and fraud (Nuwagaba, 2015). To overcome this,there has been an improvement in the uptime of the service providers through scheduled maintenance, the use of biometrics to ensure identity protection and authorisation of users. However, Uganda still experiences a series of challenges in internet banking. One major standout has been fraudulent transactions, leading to heavy losses for users (Nuwagaba, 2015).It has been studied that financial fraud, which severely affects the economy, has led to over $500 billion globally in losses realised by 2020 (Morgan, 2021). In Uganda alone, up to $11 billion was lost in internet fraud in 2019 and a further $20 billion worth of transactions challenged that were bank transfers (Kafeero, 2020).

Therefore, this research proposes using machine learning to overcome fraud challenges in internet banking in Uganda. Studies have shown that machine learning models have previouslybeen adopted to overcome financial fraud, including supervised and unsupervised approaches.

Problem Statement

There is a high level of financial fraud experienced among customers, despite the high adoptionof internet banking services in different countries. As of August 2018, up to 90% of financial fraud was experienced in the banking sector globally, with up to 18.28 billion dollars lost in providing financial services through internet banking (Syniavska et al., 2019). Because of this,as of 2019, growth in the use of banking was registered to have been stunted, falling from 3.8% in 2016 growth rate to 2.7 (Magaji, 2020). To overcome this, financial institutions have adoptednetwork security strategies, improved authentication strategies to ensure verification and validation strategies for account holders and made cross-institutional and cross-border bankingservices available at lower costs. However, there is still a high level of fraud among Uganda'sfinancial institutions. This is likely due to weak identity verification mechanisms, weakintrusion detection systems, crime from within the banking sector or weak coordination mechanisms among banks to detect fraud (Nyakarimi et al., 2020). Therefore, this research proposes developing and adopting a robust machine learning model in online payment systemsthat use previous banking data from a series of online payment transactions among financial institutions in Uganda to detect possible fraudulent transactions during online payments. This model is viewed as a strategy likely to reduce financial fraud during online payments and increase the general public's confidence in online banking systems.

Solution statement:

The study proposes adopting machine learning models to overcome fraud in online banking through timely detection, based on observed variables of both sending and recipient accounts to correctly classify fraudulent from non-fraudulent transactions (Huang, 2022). Through data mining, machine-learning strategies have been shown to successfully detect financial fraud (Awoyemi et al., 2017). Banking data is predominantly skewed, with the majority of it being non-fraudulent. Therefore, the study proposes using resampling techniques: under- sampling, oversampling, or combining both to generate synthetic samples to train a machine learning model that can enable financial institutions to detect fraud.

Different machine learning models will then be trained, with consideration for the use of grid search. It will be used to identify the best possible parameters for each model. Upon selection of the preferred model, a pipeline will be developed. The researcher proposes using an AWS S3 bucket to store data collected from a weblink as the data source. The Sagemaker pipeline will be developed to retrieve this data from the AWS- S3 bucket, pre-processed, and stored inthe feature store. Hyperparameter tuning and training for the model with the data will then be done. An endpoint will then be created, from which the model will be accessible for use throughSagemaker SDK or Amazon services like Lambda and API gateway. A model monitor will becreated that will constantly monitor the f1 score of the model, to ensure it is rectified to give absolute predictions on fraudulent transactions, and improved when the performance falls below the recommended scores.

Triggers in the pipeline will be developed at different stages to ensure constant monitoring at different stages. A manual trigger will be implemented to re-execute the pipeline. The trigger takes pipeline parameters that can be used in different pipeline steps. For example, values to retrain the model, model monitoring, etc.

Datasets and Inputs

The study intends to utilise data from 6.3 million banking records to develop, train and evaluatedifferent machine learning models based on identified performance measurement metrics in correctly detecting fraud from the dataset. The data to be used is accessible on Kaggle.com (https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset).

The dataset is comprised of 10 columns, with 6362620, 635447 (99.83%) as non-fraudulent and only 8213 (0.13%) as the fraudulent transactions.

Table 1: Dataset columns and their descriptions

No.	Column name	Description of column
1	step	represents a unit of time where 1 step equals 1 hour
2	type	type of online transaction
3	amount	the amount of the transaction
4	nameOrig	customer starting the transaction
5	oldbalanceOrg	balance before the transaction
6	newbalanceOrig	balance after the transaction
7	nameDest	recipient of the transaction
8	oldbalanceDest	initial balance of recipient before the transaction
9	newbalanceDest	the new balance of recipient after the transaction
10	isFraud	the new balance of recipient after the transaction

Benchmark Model

Different scholars have previously tried to resolve the fraud detection problem, using a series of algorithms, both ensemble and base models. Valavan & Rita (2023) developed a machine learning model to detect fraudulent transactions using machine learning for financial institutions in India. The scholars used decision trees, random forests, linear regression, and gradient-boosting methods on pre-existing data. They used precision, recall, F1, and ROC as metrics to measure the models' performances (Valavan & Rita, 2023). Perantalu and Bhargav Kiran (2017), in their study for credit card fraud detection, utilised logistic regression and decision trees as classifiers for predictive modelling of credit card fraud detection, utilising information gain to select the attributes and accuracy as the evaluation metric. However, the gap identified is that for data mining problems, especially with class imbalance, F1 score is a preferred metric to recommend for fraud detection problems. Other scholars, Perantalu and

Bhargav Kiran (2017), upon using logistic regression and decision trees, discovered that the Decision Trees performed better than Logistic Regression in detecting fraudulent transactions. This is likely because logistic regression is better used for linear data as opposed to the data that does not follow a linear characteristic. Shakya (2018) also used Random Forest, Decision trees and XGBoost to predict fraud, however, the Random Forest performed significantly better than the two models. He attributed the power of the ensemble method of the random forest in predicting fraudulent transactions. In either case, the scholars used Synthetic Minority Over-Sampling Technique (SMOTE) to solve the class imbalance problem – an oversampling technique.

Therefore, for this study, the researcher proposes utilising SMOTE, TOMEK links, and a combination of the two resampling techniques to establish their effect on the performance of the Random Forest, Decision Trees, XG boost and Random Forest models. The results from the study will be compared with those from Megdad et al (2020), as the study conducted utilised the same dataset.

Evaluation Metrics

The researcher proposes using recall score as a metric to measure the performance of the models, in comparison with those from Megdad et al (2020). The recall values for the minority class for both ensemble and base models will be compared with those from Megdad et al (2020).

Objectives of the study.

i. To develop two ensemble and two base models to predict fraudulent transactions from a heavily skewed dataset.

ii. To establish the effect of resampling on the performance of machine learning models

iii. To select the best model from the selected models.

iv. To implement the chosen model in a pipeline using AWS SageMaker.

Methodology

This section explains the procedure of how the study’s objectives were met.

Exploratory Data Analysis

The dataset was explored upon importation of the CSV file into Jupyter Notebook. All columns were checked for any missing values or duplicated values, and the data types of each of those columns was identified. Below are the results from this analysis. Also, the number of rows within the dataset was identified.

Table 1: Description of columns in the dataset

Column	**Number of missing values**	**Column Data Type**
Step	0	int
Type	0	int
amount	0	Float
nameOrig	0	Char
oldbalanceOrg	0	Float
newbalanceOrig	0	Float
nameDest	0	Char
oldbalanceDest	0	Float
newbalanceDest	0	Float
isFraud	0	int

When this was established, the type of variable for each column was determined. This was done so that the researcher could determine an appropriate method of analysing each column, and how to establish its relationship with the outcome variable (isFraud). It was established that columns, type, nameOrig, nameDest, and isFraud were determined as the categorical columns.

The unique values for each of these columns was identified, and a histogram plot was generated to visualise the distribution. In addition, the distribution of these columns with the outcome variable was also established. Below are some of the histograms developed from the exploratory data analysis process.

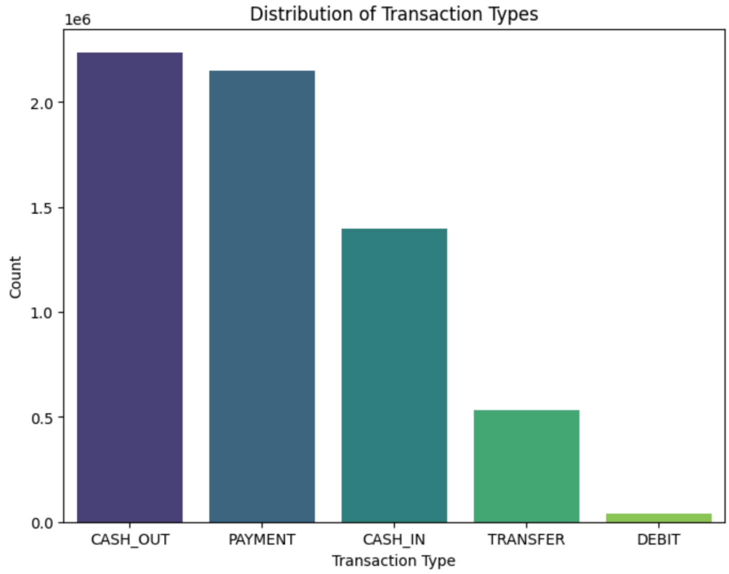

Figure 1: Distribution of Transaction Type

The fraudulent status column was also plotted to visualise the dataset's nature, revealing that the dataset was heavily skewed.

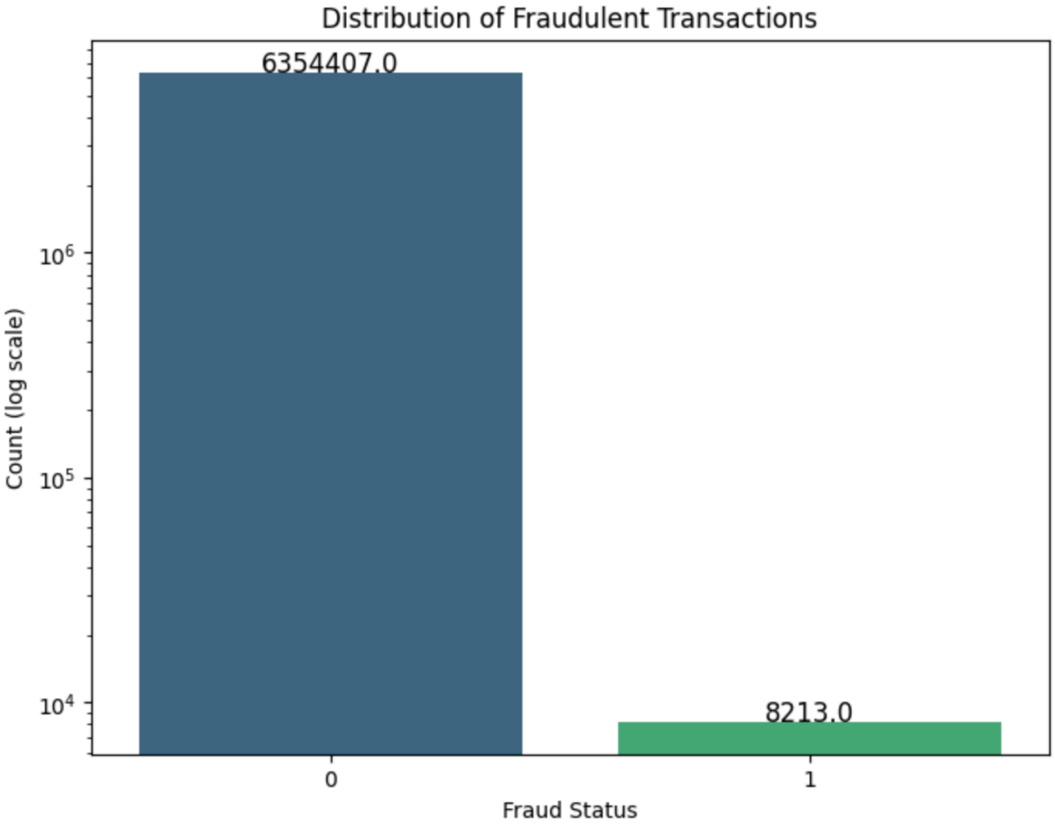

Figure 2: Visualisation of the fraud status of the dataset

This distribution of the types with the outcome variable was also determined to identify the type of the dataset where transactions were fraudulent.

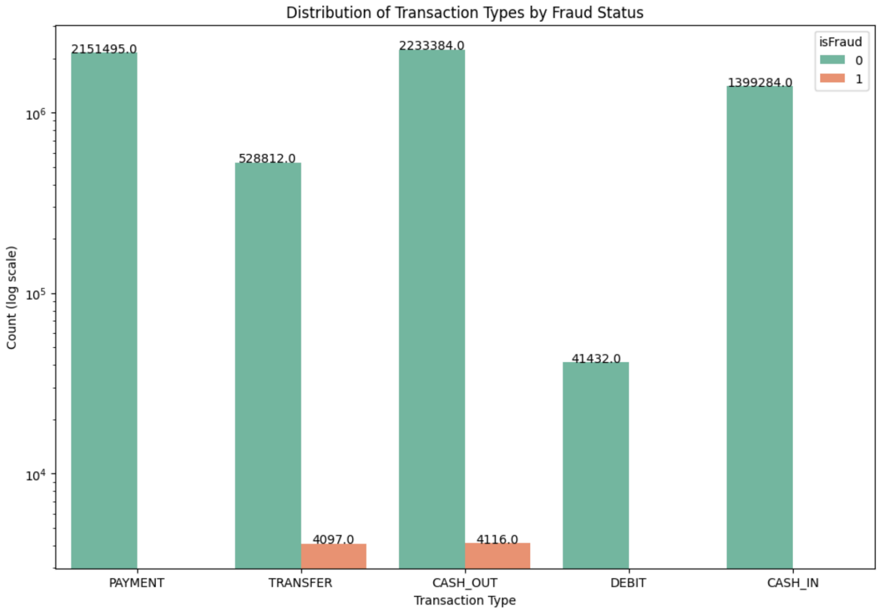

Figure 3: Distribution of fraud by type

This procedure was repeated for the different categorical columns, and, results for this, guided the process of preprocessing.

For the continuous columns, their correlation, linearity, and distribution of each of these columns were established. The continuous columns included step, amount, oldbalanceOrg newbalanceOrig, oldbalanceDest, and newbalanceDest.

Checking for linearity was to guide feature engineering and identify appropriate techniques to estimate the relationship between variables. It was established that all the continuous columns were non-linear from the dataset. In addition, they were not normally distributed.

The columns that had a high correlation were identified and one of those was dropped. This was because of multicollinearity, a condition that could cause the model to overfit.

Pre-processing of variables

Different techniques were used to pre-process the different variables in the dataset. This was done to ready the data to train the model and as well ensure the model is not over or under fitted. Different data pre-processing techniques were employed for the different columns depending on the insight gathered from the data columns during the EDA stage.

The table below summarises actions to the different columns.

Table 2: Preprocessing steps for the different variables

VARIABLE TYPE	VARIABLE	PREPROCESSING ACTION
CATEGORICAL	Type	In this, the researcher identified that the rows payment, debit and cash-in of the type column did not have any fraudulent transactions. They were all renamed to other. Using a label encoder, transfer, cashout and other were renamed to integers 1, 2 and 3 respectively.
	nameOrig and nameDest	It was identified that there are about 6.3m origin accounts transacting with about 2.7m destination accounts. From these accounts, all business accounts did not register a fraudulent transaction. Therefore, all rows that had customer-to-business transactions were dropped.
	Step	For the step column, since they were total hours, they were converted into days of the week. The reference date was 1/01/2023. The hours were converted into days. The days were then converted into numeric using an ordinal encoder.
CONTINUOUS COLUMNS	amount	Linearity and normality tests revealed skewed behavior of these variables and non-linearity. Spearman's correlation coefficient was used to measure the correlation between the variables. Columns oldbalanceOrg, newbalanceOrg, oldbalanceDest, and newbalanceDest had a high correlation. Only two were therefore selected for the model, alongside the column amount that had a weak correlation with any of those columns.
	oldbalanceOrg	
	newbalanceOrg	
	oldbalanceDest	
	newbalanceDest	

The data frame was then split into the independent variable X and the dependent variable y. Different models were then trained with the dataset.

In the first instance, the dataset was split into training and testing. The continuous columns in the training set were scaled using a robust scaler, and the same for the testing set. Four models, two being ensemble and two being base models were trained and then the prediction was got. The same procedure was repeated, but, this time, with over sampling using SMOTE and using a combination of SMOTE and TOMEK links.

Results

This section demonstrates the findings of the study's objectives.

The table below shows the results from these procedures for the minority class.

Table 3: Results for minority class for the different models

MODEL TYPES	**MODELS**	**Re-sampling Type**	**PERFORMANCES**		
			Precision	**Recall**	**F1**
Ensemble Models	XG boost	No sampling	0.889851	0.443827	0.592257
		SMOTE	0.998418	0.998341	0.998379
		SMOTETOMEK	0.998628	0.998850	0.998739
	Random Forest	No sampling	0.865911	0.566049	0.684584
		SMOTE	0.999629	0.991231	0.995412
		SMOTETOMEK	0.9994	0.8398	0.9127
Base Models	Decision Trees	No sampling	0.6931	0.6525	0.6722
		SMOTE	0.9995	0.9647	0.9818
		SMOTETOMEK	0.9995	0.9792	0.9892
	KNN	No sampling	0.8335	0.4667	0.5983
		SMOTE	0.9979	0.9811	0.9894
		SMOTETOMEK	0.9980	0.9815	0.9897

The results for the minority class for each of the models were collected. It was observed that the models were random in their prediction when the resampling techniques were not implemented. Particularly, their recall scores were low, averaging about 50%.

Using over-sampling (SMOTE) and a combination of over-sampling and under-sampling (SMOTE + TOMEK links), the models' performance significantly improved compared to its performance without resampling. This is likely attributed to the models being trained with more samples of the minority class, thereby gathering more insight into their distribution.

Analysis shows that there was a very slight difference in the performance of the model performance between using SMOTE and using SMOTE + TOMEK links.

The XG boost model performed best at predicting fraudulent cases after oversampling, with the highest F1 and recall values compared to all the other models.

Model implementation and deployment

Upon development and selection of the preferred model, a pipeline was developed. For every stage of the pipeline, different libraries were utilised. Technologies including Lambda and

Amazon S3 bucket, alongside sklearn libraries were used to implement, deploy, and make the model scalable for use.

At the processing stage, the pipeline was configured to pre-process the raw data of the identified columns. It involved steps of identifying any missing columns, duplicated rows, and data augmentation, in preparation for model training. In this stage, sklearn script processor was used to pull data from the S3 bucket, pre-processed, feature engineered as done during the model selection process. The pipeline was then configured to send the processed data back to the S3 bucket. A feature store was also created in order to monitor the features to be used for model training.

In the training stage, the pipeline was configured to collect the pre-processed data and have it fed into the identified machine learning model. The pipeline then was configured to save the trained model artifacts back into the S3 bucket.

To evaluate the model, the recall value for the minority class (Fraudulent cases) was used as a metric to estimate model performance. The saved metrics of the model were used to conduct predictions on unseen data whose results were put in a validation report and saved back into the S3 bucket.

To set the accuracy condition, it determined whether or not to run the pipeline. The decision node was set to a recall value of 0.99 as a performance threshold. The pipeline was configured to fail at any score below this value. The sklearn repack model was used to format the model for deployment, and reused in a production environment. The create model was also implemented to package the model for deployment. This involved setting up configurations for model serving, like endpoint configuration, using Sagemaker.

The model was then registered in the model registry, which is a repository where models are stored, versioned, and managed. Registration typically includes tagging with metadata for traceability and governance. We register packaged models along with other data like model bias and explainability of the model in sagemaker model registry where it will be approved manually. The AWS lambda functions were used to trigger the pipeline at different stages automatically. These included starting the pipeline execution, deploying model endpoint, and performing inference on the deployed model.

Below is the architectural diagram for the pipeline implementation.

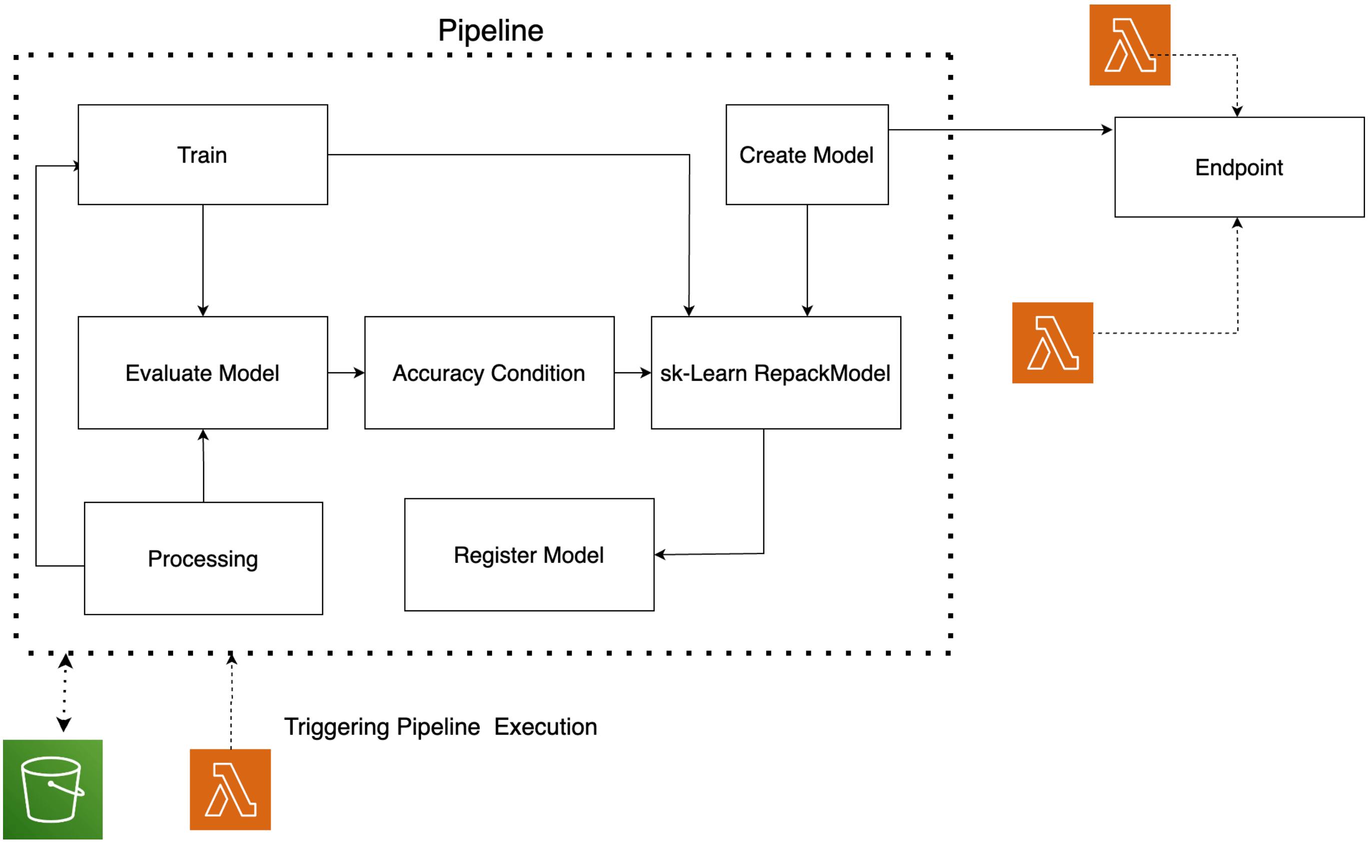

Figure 4: Pipeline implementation for the best model

Discussion

Two ensemble techniques - Random Forest Classifier and XG boost – were used to meet the study's objectives. In addition, two base models - KNN and Decision trees - were also used to identify the best possible model to predict fraudulent transactions. The study used recall, precision and F1 score to compare the performances between the models. Similar studies have also used F1, recall, and precision as metrics to measure and compare models for fraud detection (Han et al., 2020; Itoo et al., 2021; Sadineni, 2020). Moreover, for the same dataset, different scholars that studied the same dataset compared the performance of a series of models with and without resampling and estimated its effect on the quality of predictions of the minority class. Megdad et al (2022) compared the performance of different machine learning models without resampling and with the use of the SMOTE technique to ensure representatives of the classes. For the random forest classifier, an algorithm also used in this study showed that SMOTE technique significantly improved its performance (Megdad et al., 2022).

However, Megdad et al (2022) utilised about 60% of the data for training, and the rest for testing and validation, which explains the slight variance in their findings. In this study, 80% of the data was used to train the data. In addition, utilising SMOTE+TOMEK links further increased the performance of the models and showed that using over-sampling techniques improves the recall of fraud detection among algorithms. In fact, the best identified model was the XG Boost classifier (Raju, 2021).

Conclusion:

From the study, it was discovered that pre-processing data significantly determines the choice of models to use. In addition, use of resampling techniques, specifically over sampling, further improves the performance of models, while preserving the amount of data for the model to train, in predicting fraudulent transactions.

In addition, AWS ML Ops pipeline streamlines the process of transforming data into a predictive model ready for production. By automating these steps within the AWS ecosystem, you can ensure that the model's deployment is both robust and aligned with best practices for machine learning operations. It also maintains a high standard of the model’s quality, improves monitoring and governance.

Recommendations:

I recommend the use of grid search and cross validation to improve the recall of the model even further. I also recommend the use of hyper parameter tuning to ensure the most effective features of the model are utilised to produce the most effective result for the model.

I also recommend future researchers to use event bridge to automatically trigger the pipeline execution for applications where new data is coming in regularly. Also, adding services like AWS code commit to aid in versioning the code for pre-processing.

REFERENCES

Ali, G., Ally Dida, M., & Elikana Sam, A. (2020). Evaluation of key security issues associated with mobile money systems in Uganda. *Information*, *11*(6), 309.

Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, 29-31 Oct. 2017). Credit card fraud detection using machine learning techniques: A comparative analysis. 2017 International Conference on Computing Networking and Informatics (ICCNI),

Han, Y., Yao, S., Wen, T., Tian, Z., Wang, C., & Gu, Z. (2020). Detection and analysis of credit card application fraud using machine learning algorithms. Journal of Physics: Conference Series,

Huang, X. S. V. J. X. (2022). Detect fraudulent transactions using machine learning with Amazon SageMaker.

Itoo, F., Meenakshi, & Singh, S. (2021). Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology*, *13*, 1503-1511.

Magaji, B. (2020). A Legal Overview of Electronic Banking in Uganda.

Megdad, M. M., Abu-Naser, S. S., & Abu-Nasser, B. S. (2022). Fraudulent financial transactions detection using machine learning.

Morgan, R. E. (2021). *Financial fraud in the United States, 2017*. US Department of Justice, Office of Justice Programs, Bureau of Justice ….

Nuwagaba, A. (2015). E-Banking Performance in Uganda: A Case Study of Bank of Uganda. *East Asian Journal of Business Economics (EAJBE)*, *3*(2), 13-20.

Nyakarimi, S. N., Kariuki, S. N., & Kariuki, P. (2020). Risk assessment and fraud prevention in banking sector.

Raju, O. (2021). Credit Card Fraud Detection Using Xgboost Classifier. *International Journal of Techno-Engineering*, *13*(III).

Sadineni, P. K. (2020). Detection of fraudulent transactions in credit card using machine learning algorithms. 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC),

Syniavska, O., Dekhtyar, N., Deyneka, O., Zhukova, T., & Syniavska, O. (2019). Security of e-banking systems: Modelling the process of counteracting e-banking fraud. SHS Web of Conferences,

Valavan, M., & Rita, S. (2023). Predictive-Analysis-based Machine Learning Model for Fraud Detection with Boosting Classifiers. *Computer Systems Science & Engineering*, *45*(1).