CREDIT CARD FRAUD DETECTION

TABLE OF CONTENT

REQUIREMENTS FOR THE MODEL

absl-py==1.4.0
psycopg2==2.9.6
pyasn1==0.5.0
pyasn1-modules==0.3.0
pycryptodome==3.10.1
pyFPE==0.10.3
pyparsing==3.1.0
python-dateutil==2.8.2
pytz==2023.3
requests==2.30.0
requests-oauthlib==1.3.1
rsa==4.9
scikit-learn==1.2.2
scikit-multiflow==0.5.3
scipy==1.10.1
six==1.16.0
sortedcontainers==2.4.0
tensorboard==2.12.3
tensorboard-data-server==0.7.0
tensorflow==2.12.0
tensorflow-estimator==2.12.0
tensorflow-io-gcs-filesystem==0.32.0
termcolor==2.3.0
threadpoolctl==3.1.0
typing_extensions==4.5.0
tzdata==2023.3
urllib3==2.0.2
Werkzeug==2.3.3
wrapt==1.14.1
zipp==3.15.0
astunparse==1.6.3
blinker==1.6.2
cachetools==5.3.0
certifi==2022.12.7
charset-normalizer==3.1.0
click==8.1.3
contourpy==1.1.0
cycler==0.11.0
docutils==0.20.1
ff3==1.0.1
Flask==2.3.2
flatbuffers==23.3.3

fonttools==4.40.0
gast==0.4.0
geohash2==1.1
google-auth==2.17.3
google-auth-oauthlib==1.0.0
google-pasta==0.2.0
grpcio==1.54.0
h5py==3.8.0
idna==3.4
importlib-metadata==6.6.0
importlib-resources==5.12.0
itsdangerous==2.1.2
jax==0.4.8
Jinja2==3.1.2
joblib==1.2.0
kafka-python==2.0.2
keras==2.12.0
kiwisolver==1.4.4
libclang==16.0.0
Markdown==3.4.3
MarkupSafe==2.1.2
matplotlib==3.7.1
ml-dtypes==0.1.0
numpy==1.23.5
oauthlib==3.2.2
opt-einsum==3.3.0
packaging==23.1
pandas==2.0.2
Pillow==9.5.0

ASSUMPTIONS

- **Incoming Data:** The incoming data is assumed to have rows structured similarly to the dataset on which the model was trained, including consistent column names and the data types.

- **Temporal Dependencies:** Assume that there are temporal dependencies in credit card transactions. In other words, fraudulent transactions may exhibit patterns over time that differ from legitimate transactions. LSTM, as a recurrent neural network (RNN), can capture these temporal dependencies effectively.

- **Localised Context:** Assume that fraudulent transactions may exhibit localised context within a certain time window or geographical area. By considering local patterns and relationships, the model can identify anomalies more accurately. This assumption aligns well with the sequential nature of LSTM.

- **Real-Time Environment:** The model assumes that the real-time fraud detection system is set up and operational. It expects a reliable and consistent stream of data flowing into the system in real-time.

- **Diverse Feature Set:** Assume that the dataset includes various features related to credit card transactions, such as transaction amount, time of the day, merchant category, customer's location, etc. By considering a diverse set of features, the model can learn more nuanced patterns and improve its fraud detection capabilities.

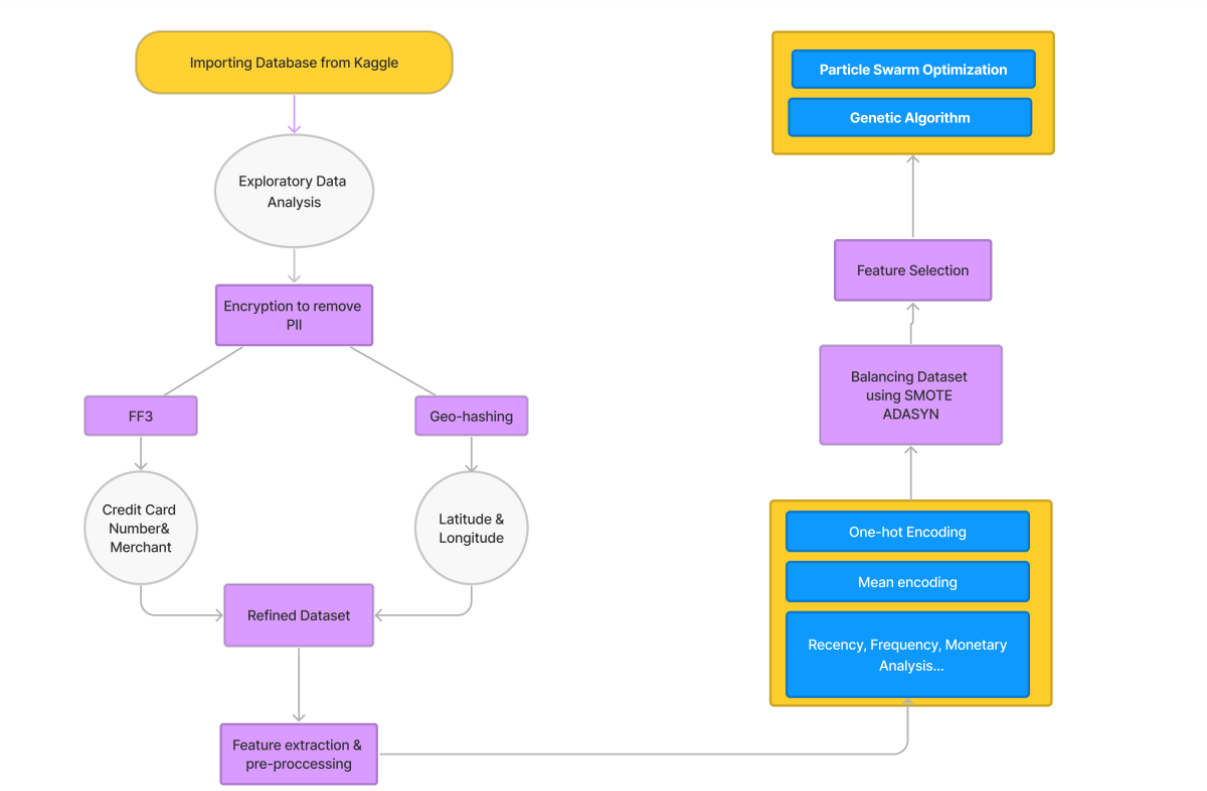
ABOUT THE DATASET

We have used a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants.

This dataset was generated using Sparkov Data Generation | Github tool created by Brandon Harris. This simulation was run for the duration - 1 Jan 2019 to 31 Dec 2020. The files were combined and converted into a standard format.

PRE-PROCESSING

Pre Processing is an essential step in credit card fraud detection to gain a better understanding of the data and uncover patterns, anomalies, or trends that may indicate fraudulent activities.



ENCRYPTION

To ensure data security and privacy, sensitive information like credit card numbers (cc_num), merchant details, and latitude-longitude coordinates are encrypted using encryption techniques. This helps protect confidential information while retaining its usability for analysis.

Encryption plays a critical role in safeguarding Personally Identifiable Information (PII) and Personal Sensitive Information (PSI) within various systems and models.

We have used FFP and Geo Hashing techniques.

1) FF3

We have used FF3 to encrypt credit card numbers while preserving their original format and length. It is a format-preserving encryption scheme that allows for reversible encryption of data within a specified format. The algorithm ensures that the encrypted output maintains

the same format as the original credit card number, including the same number of digits. This is important because many systems and processes rely on the standard format of credit card numbers.

2) GeoHashing

Geo-hashing is a technique used to encode and index geographical coordinates, such as latitude and longitude, into a string format. It allows for efficient spatial indexing and proximity-based searching. We have converted latitude and longitude coordinates into a string format that retains spatial information. This enables efficient indexing and searching of geographical data based on proximity

Understanding Customer Patterns:

RFM (Recency, Frequency, Monetary) Measure

RFM (Recency, Frequency, Monetary) is performed to derive meaningful features from the transaction data. RFM metrics capture the recency of transactions, the frequency of transactions, and the monetary value of transactions, along with the risk and credit score of merchants and categories.We have used this to gain a better understanding of customer behaviour and identify segments with different characteristics. It provides insights into customer value and engagement by analysing three key dimensions: recency, frequency, and monetary value.

Recency: Recency refers to the time elapsed since the cardholder's most recent transaction. Monitoring recency helps identify any sudden or unusual changes in transaction behaviour. Fraudsters often exhibit irregular transaction patterns, such as a sudden increase in activity or transactions made outside the cardholder's usual behaviour. By considering recency, fraud detection systems can detect potentially fraudulent activities in real time.

Frequency: Frequency measures the number of transactions conducted by a cardholder within a specific time period. Analysing transaction frequency helps establish a baseline for normal behaviour. Unusual spikes or irregular patterns in transaction frequency can indicate potential fraudulent activity. For instance, multiple

transactions within a short duration or a sudden surge in transaction frequency may signal unauthorised card usage or account takeover.

Monetary: Monetary refers to the value or amount associated with each transaction. Analysing the monetary aspect helps identify unusual transaction amounts compared to a cardholder's typical spending behaviour. Fraudsters often attempt to exploit stolen card details by making large or irregular transactions that deviate from the cardholder's normal spending patterns. Monitoring monetary values helps flag potentially fraudulent transactions based on significant deviations from expected spending amounts.

Additionally, RFM enables customised risk scoring tailored to individual cardholders' transaction behaviour, enhancing the effectiveness of fraud detection systems.

Benefits of RFM :

- Comprehensive Risk Assessment
- Early Fraud Detection
- Enhanced Accuracy
- Customised Risk Score
- Pattern Recognition

Sampling of Dataset

Credit card fraud is an **imbalanced class problem**, where the number of fraudulent transactions is significantly lower than legitimate transactions, therefore ADASYN (Adaptive Synthetic Sampling) is used to generate synthetic instances of the minority class (fraudulent transactions) based on their feature distributions. This helps to address the class imbalance issue and provides a more representative dataset for training the model.

Reasons to use ADASYN:

- **Retaining natural characteristics of minority class**: ADASYN focuses on generating synthetic samples that are closer to the natural distribution of the minority class. By considering the local density and adjusting the synthetic sample generation accordingly, ADASYN aims to preserve the intrinsic characteristics and patterns of the minority class. This can lead to better generalisation and improved performance in detecting credit card fraud instances.

- **Reduced risk of overfitting**: Oversampling techniques can introduce a risk of overfitting, where the model becomes too specialised in the synthetic examples and may not generalise well to real-world scenarios. ADASYN's adaptive sampling strategy helps mitigate this risk by generating synthetic samples based on the difficulty of learning, effectively increasing the diversity of the synthetic samples. This promotes better generalisation and reduces the likelihood of overfitting.

- **Handling class imbalance effectively**: Credit card fraud datasets often suffer from severe class imbalance, where the number of fraudulent transactions is significantly smaller than legitimate transactions. ADASYN (Adaptive Synthetic Sampling) is designed to address this issue by generating synthetic examples for the minority class based on their difficulty of learning. It focuses on areas with fewer examples, increasing the generation of synthetic samples in those regions. This adaptive sampling approach can help to alleviate the imbalance problem more effectively.

- **Dealing with noisy and overlapping samples**: In credit card fraud detection, the minority class (fraudulent transactions) can be highly diverse and overlap with legitimate transactions, making it challenging to generate meaningful synthetic samples. ADASYN considers the density distribution of minority samples and generates synthetic samples in proportion to their local density. This approach helps to mitigate the impact of noisy samples and avoids creating synthetic examples in less representative regions.

FEATURE SELECTION TECHNIQUES IMPLEMENTED

Weighted feature selection techniques are used to determine the importance or relevance of features in a dataset by assigning weights to them. These weights reflect the impact of each feature on the target variable or the overall model performance.

Two feature selection are implemented:

- Genetic Algorithm
- Particle Swarm Optimization

PSO and GA are used to optimise the selection process by evaluating different subsets of features and finding the most informative combination. These techniques consider the performance of the model and search for the best feature subset that maximises accuracy and minimises overfitting.

1) Genetic Algorithm

Genetic algorithms are well-suited for feature selection, which is crucial in credit card fraud detection. GAs can automatically search through a large feature space and identify the most informative subset of features for fraud detection. By selecting relevant features, GAs can improve the model's performance, reduce overfitting, and enhance interpretability.

Advantages of Genetic Algorithm:

- **Feature interaction and redundancy**: GAs can capture the interactions between features and identify redundant features that provide similar information. By eliminating redundant and irrelevant features, GAs can improve model performance and reduce overfitting.
- **Optimization of model hyperparameters**: GAs can optimise the hyperparameters of the fraud detection model, such as the learning rate, regularisation parameters, or ensemble weights. By searching the hyperparameter space, GAs can find optimal configurations that maximise the model's performance. This optimisation process can lead to improved accuracy and robustness in fraud detection.

- **Handling complex and nonlinear relationships**: Credit card fraud detection involves identifying intricate patterns and relationships among various features. GAs can capture complex and nonlinear relationships by iteratively combining different features and evaluating their impact on the model's performance. This flexibility allows GAs to handle complex fraud detection tasks that may not be easily addressed by other techniques.
- **Handling multicollinearity:** GAs can address the issue of multicollinearity, where features are highly correlated with each other. By selecting a subset of features that are less correlated, GAs can improve the stability and interpretability of the resulting models.

2) PSO

PSO is a metaheuristic optimization algorithm that aims to find the optimal solution by iteratively searching the solution space. It has the ability to explore a wide range of feature combinations and search for the global optimum. This global optimization capability can be beneficial in credit card fraud detection, as it allows for a comprehensive search for the most informative feature subset.

Advantages of PSO:

- **Parallelizable and efficient**: PSO is a parallelizable algorithm that can be efficiently implemented. It can explore the solution space using multiple particles simultaneously, making it suitable for large-scale credit card fraud datasets with a high number of features. The parallel nature of PSO allows for efficient utilisation of computational resources and enables faster feature selection compared to exhaustive search techniques.

- **Balancing accuracy and complexity**: PSO can strike a balance between the accuracy of the fraud detection model and the complexity of the feature subset. By optimising the feature selection process, PSO can identify a subset of features that maximises the performance of the model while minimising the number of features used. This helps to reduce computational complexity, improve interpretability, and mitigate the risk of overfitting.

- **Adaptability and flexibility**: PSO can adapt to changing fraud patterns over time. As fraud patterns evolve, the relevance and importance of different features may change. PSO can reevaluate and update the feature subset based on the evolving fraud patterns, ensuring that the model remains effective and up to date

ABOUT THE PROPOSED MODEL

LSTM

LSTM layers are a type of recurrent neural network layer that can capture long-term dependencies and patterns in sequential data.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization (Normalizatio  (None, 1, 26)            53
 n)

 lstm (LSTM)                 (None, 1, 128)            79360

 lstm_1 (LSTM)               (None, 1, 128)            131584

 global_average_pooling1d (G  (None, 128)              0
 lobalAveragePooling1D)

 dropout (Dropout)           (None, 128)               0

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 211,126
Trainable params: 211,073
Non-trainable params: 53
```

- We have applied a normalisation layer at the beginning of the model to normalise the input data. Normalisation typically involves scaling the input values to a standard range, such as between 0 and 1 or -1 and 1. This step helps in stabilising the training process and ensures that the input features are within a similar range.
- Then, two LSTM layers are added to allow a more complex representation of the data, enabling the model to learn hierarchical and intricate relationships within the sequences.
- The last layer is, global layer, also known as a flatten layer, is used to convert the output of the last LSTM layer into a one-dimensional vector.
- It reshapes the output from a 3D tensor (sequence length x LSTM units) to a 2D tensor (sequence length x LSTM units) or a 1D tensor (sequence length x LSTM units) depending on the specific implementation. This step is typically done to prepare the data for subsequent layers, such as fully connected layers or the output layer.

Advantages of LSTM:

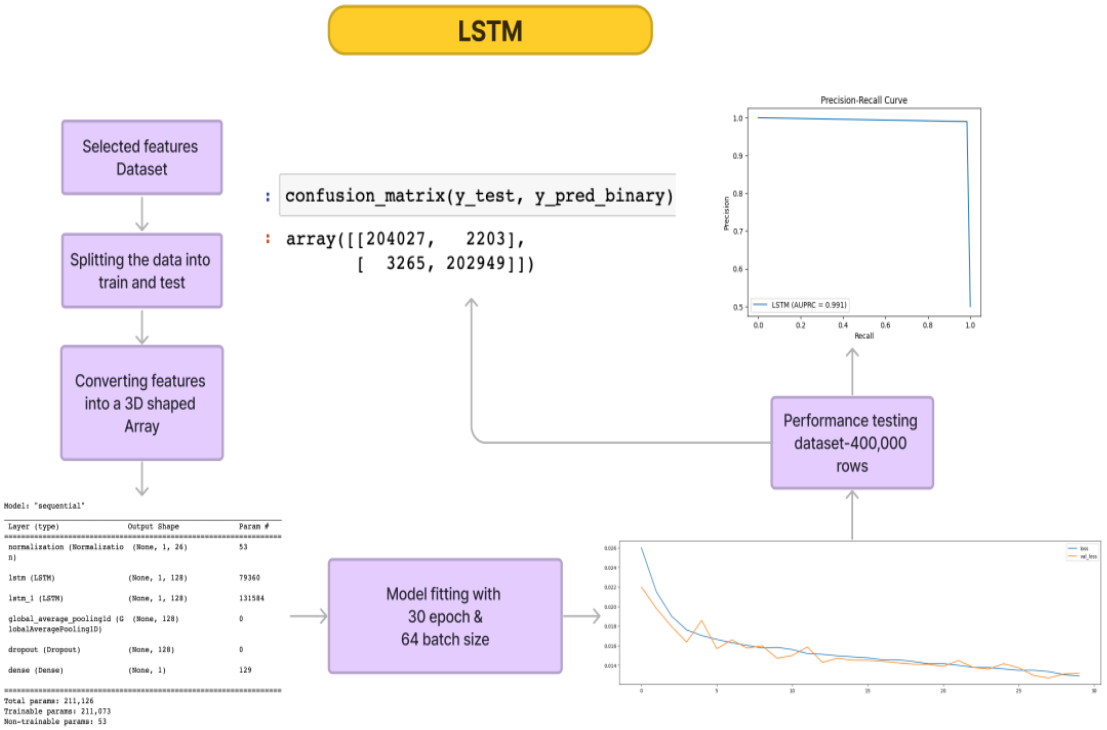
- **Handling temporal dependencies**: Credit card transactions often exhibit temporal dependencies, where the order and timing of transactions can provide important context. LSTM layers are specifically designed to capture long-term dependencies in sequential data. They have a memory cell that can retain information over long periods, allowing them to model and learn patterns in sequences effectively.

- **Capturing context and time-sensitive patterns**: Fraudulent activities often involve intricate patterns that can span multiple transactions or occur over a specific period. LSTM layers can capture contextual information and identify time-sensitive patterns by learning from past transactions in the sequence. They can detect anomalies or deviations from expected behaviours, which is crucial in fraud detection.

- **Modelling variable-length sequences**: The length of credit card transactions might fluctuate, and LSTM layers can handle variable-length input sequences. They can analyse and learn from sequences of varying lengths without the need for fixed-size inputs, which is a significant benefit over other neural network topologies.

- **Sequential feature extraction**: LSTM layers can extract useful features from sequential data automatically. They can learn to recognise crucial patterns and correlations within transaction sequences that are difficult to capture using other neural network architectures. This feature extraction capacity improves the overall performance of fraud detection.

- **Dealing with skewed data**: Credit card fraud datasets are often skewed, with the bulk of transactions being valid and only a small fraction being fraudulent. By learning from both positive (fraudulent) and negative (legitimate) instances, LSTM layers can manage unbalanced data. Their ability to retain information from prior time steps aids in the capture of unusual events and the prediction of accurate outcomes.



Adaptive Random Forest

Adaptive Random Forest is applied to handle evolving fraud patterns and maintain the model's performance as new data is encountered.The model is trained using the selected features and the oversampled dataset.It leverages the diversity of individual decision trees to enhance fraud detection accuracy.We created an instance of the AdaptiveRandomForestClassifier from the skmultiflow.meta module. The ARF classifier is configured with 200 decision trees (n_estimators) and a lambda value . To evaluate the performance of the ARF classifier, we utilise the EvaluatePrequential class from skmultiflow.evaluation. It performs a prequential evaluation, which combines elements of both pre-training and incremental learning

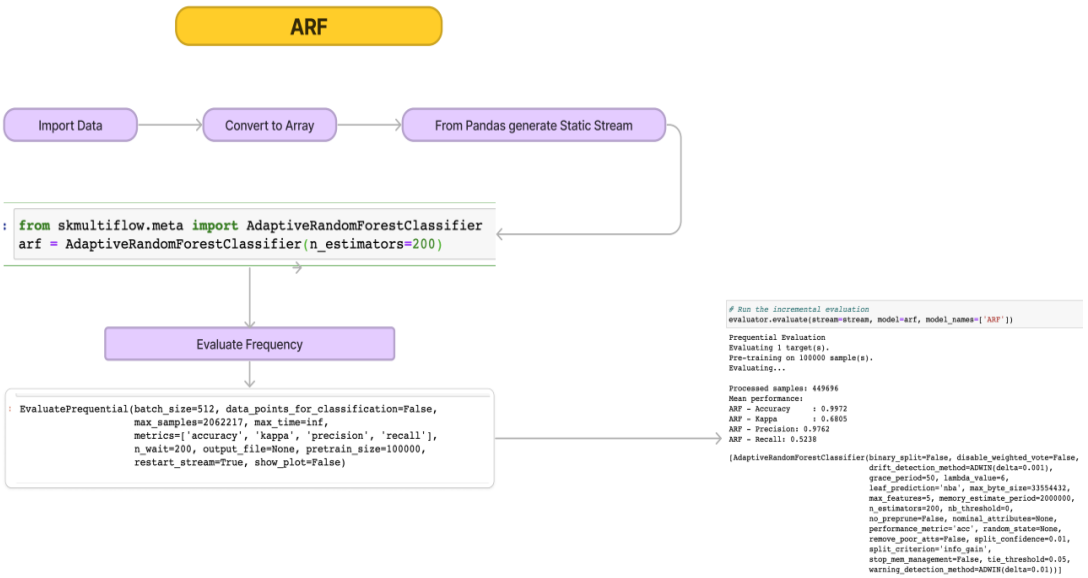
Advantages of Adaptive Random Forest

- **Detection of concept drift**: Concept drift is the phenomenon in which the underlying data distribution changes over time. Fraudsters regularly change their strategies in credit card fraud detection, resulting in developing patterns in fraudulent operations. By monitoring the performance of its base classifiers, ARF incorporates a mechanism for detecting and dealing with idea drift. When idea drift is found, new classifiers are developed to adapt to changing patterns, ensuring that the model remains current and effective in detecting fraud.

- **Efficiency:** ARF is intended to be efficient and scalable, even for big datasets. It uses randomization approaches for feature selection and sampling, lowering computer complexity while retaining prediction accuracy. This can be useful when dealing with big amounts of credit card transaction data.

- **Ensemble-Based Approach:** ARF is an ensemble method for making predictions that incorporates numerous base classifiers. Each base classifier is trained on a separate subset of features and/or data, which improves the model's robustness and lowers the danger of overfitting. When compared to individual classifiers, the combination of many classifiers provides for higher generalisation and overall performance.

- **Imbalance Data Handling:** ARF employs a dynamic feature bagging technique that adapts to the data's imbalance. It chooses distinct feature subsets for different base classifiers, lowering the risk of favouring the majority class and boosting performance on minority class cases.



Combined Predictions: Weighted Votinng

The LSTM and ARF models are combined to leverage their complementary strengths. This fusion approach integrates the predictions from both models, effectively capturing different aspects of fraud patterns and improving the overall detection performance. By following these steps, the credit card fraud detection model employs encryption for data security, calculates RFM metrics, applies to oversample with ADASYN, performs feature selection

using PSO and GA, utilises LSTM and ARF models, and combines them to create a comprehensive fraud detection system.

Advantages of the Combined Model Approach

- **Improved Decision Making:** The combined weights model enables you to make decisions based on a consensus of both ARF and LSTM predictions. By assigning appropriate weights to each model's predictions, you can prioritise certain aspects (e.g., accuracy, speed, interpretability) or give more weight to the model that performs better on specific types of fraud patterns. This flexibility allows you to optimise the decision-making process based on your specific requirements and priorities.

- **Handling different types of features**: Credit card fraud detection often involves various types of features, including static customer information, transaction metadata, and sequential transaction data. ARF can handle static and metadata features effectively, while LSTM can capture sequential patterns. By integrating ARF and LSTM, you can leverage their respective strengths to handle different types of features and extract the most relevant information for fraud detection.

- **Adaptive learning**: Concept drift is a common challenge in credit card fraud detection, as fraud patterns evolve over time. ARF's adaptive mechanism allows it to detect and adapt to concept drift, updating its base classifiers accordingly. LSTM, with its ability to capture temporal dependencies, can learn from historical transaction sequences and adapt to changing fraud patterns. The combined weights model can effectively adapt to evolving fraud scenarios and maintain detection accuracy over time.

- **Complementary strengths**: ARF and LSTM have different strengths in capturing patterns and detecting anomalies in credit card transactions. ARF is effective in handling imbalanced data and concept drift, while LSTM excels at capturing sequential dependencies and temporal patterns. By combining these two models, you can leverage their complementary strengths and enhance the overall fraud detection performance.

PERFORMANCE METRICS

When dealing with imbalanced datasets, the number of instances in one class significantly outweighs the other, traditional performance metrics like accuracy can be misleading. Therefore, it is important to use evaluation metrics that provide a more comprehensive understanding of the model's performance in such scenarios. Here are some commonly used performance metrics for our dataset:

Precision: Precision is the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive. Precision is particularly useful in imbalanced datasets because it focuses on the accuracy of positive predictions. It helps assess the model's ability to avoid false positives, which is crucial in scenarios where misclassifying the minority class has significant consequences, such as in credit card fraud detection.

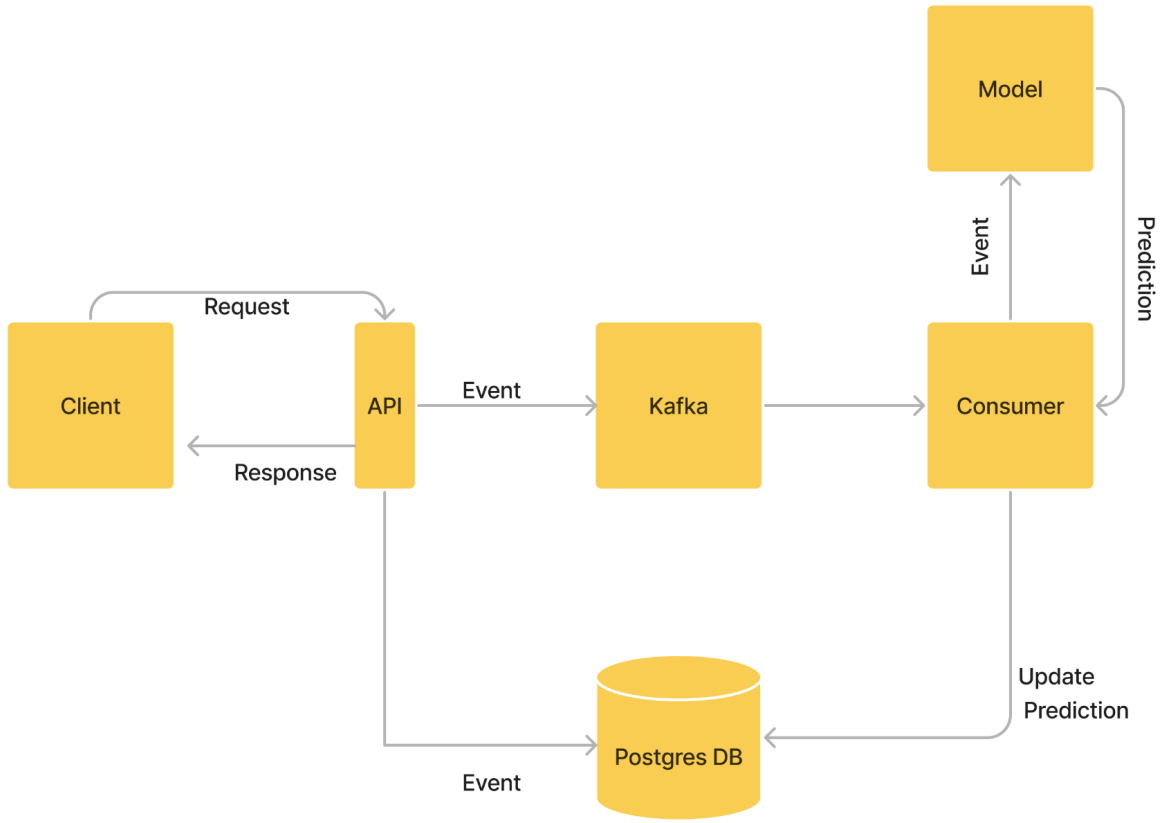
Area Under the Precision-Recall Curve (AUPRC): The Precision-Recall curve plots precision against recall at different classification thresholds. AUPRC summarises the overall performance of the model across different threshold values. AUPRC is commonly used in imbalanced datasets because it captures the trade-off between precision and recall, comprehensively evaluating the model's performance across different operating points. It is especially useful when the class distribution is heavily imbalanced.

Recall (Sensitivity or True Positive Rate): Recall is the proportion of correctly predicted positive instances (true positives) out of all actual positive instances. It measures the model's ability to capture positive instances and avoid false negatives. In imbalanced datasets, recall is vital as it indicates the model's sensitivity in detecting the minority class. It helps evaluate the model's ability to identify instances of the minority class, such as detecting fraudulent transactions in credit card fraud detection.

WHY ACCURACY IS NOT CONSIDERED IN OUR APPROACH

In imbalanced datasets, the majority class typically dominates the overall distribution, while the minority class represents a small portion of the data. As a result, a model that predicts only the majority class for all instances can achieve a high accuracy simply by capitalising on the imbalanced class distribution. This can lead to a misleadingly high accuracy score while providing little or no value in detecting the minority class. Accuracy treats all classes equally and does not differentiate between correct predictions of the majority class and correct predictions of the minority class.

SIMULATING REAL TIME ENVIRONMENT



- **Flask API:** Our robust Flask API serves as the communication interface for real-time data transmission. It efficiently sends data to the Kafka producer, ensuring seamless integration between components.

- **Kafka Producer**: The Kafka producer receives the data from the Flask API and publishes events to the Kafka topics. In addition to publishing events, the producer also adds the received events directly to the **PostgreSQL** database. This allows for immediate persistence of the event data.

- **Kafka Consumer:** Our consumer component subscribes to the Kafka topics, consuming the events generated by the producer in real time. Upon receiving an event, the consumer performs various preprocessing tasks using machine learning-based techniques. This ensures that the data is transformed into the appropriate format required for accurate predictions.

- **Machine Learning Predictions:** After preprocessing, our consumer leverages our trained machine learning models to make accurate predictions based on the specific requirements of our system, such as fraud detection or anomaly identification. These predictions are generated in real time, ensuring timely insights.

- **Storing Event Data:** The final predicted outputs, along with the corresponding event data, are associated and stored in our **PostgreSQL** database. This enables us to maintain a comprehensive record of all events and their respective predictions. Furthermore, the Kafka producer's direct addition of events to the database ensures immediate storage.

CONCLUSION

The proposed Model is able to solve three major problems in credit-card fraud detection problem:

- Class Imbalance- Imbalanced data can lead to biased machine learning models that struggle to detect the minority class (fraudulent transactions). ADASYN used to handle this problem.
- Concept Drift - Adapting to patterns of fraudulent activities over time, which can impact the overall effectiveness of fraud detection model.
- Verification Latency- It encompasses the duration required for fraud detection analysis, impacting user experience and necessitating efficient fraud mitigation strategies for timely verification and response. The proposed approach solves the problem by giving accurate predictions within seconds.