Time series analysis projecT

reporT

**Fraud Detection In Banks Using ARIMA and**

**ML/Deep Learning Models**

# PROJECT OBJECTIVE AND RELEVANCE

The goal and purpose of the project are to detect fraudulent behaviour in the bank to prevent associated risks and financial fraud. Neo-banks use ML to detect fraud and malicious attempts. The fraud detection system can be built using time series techniques integrated with other banking applications, such as credit risk scoring or online banking fraud. ARIMA combined with recurrent neural networks can be suitable to work on such an application.

Credit cards have become a convenient and widely used medium for monetary transactions. Due to their increased usage, the number of frauds has exponentially shot up as well. Scammers are also evolving their tactics and it’s becoming more difficult to identify and prevent scams.

Regular or non-fraudulent transactions are consistent with time and show certain behavioural patterns. Anomalies in their data chart are usually a strong indicator of fraudulent activities. Such irregularities must be immediately tracked and tackled to hinder damages.

By developing a robust detection model, we can contribute to enhancing the security of financial institutions and reduce fraudulent activities. Moreover, this project provides an opportunity to test and apply advanced time-series modelling techniques to a real-world problem, thereby improving the predictive capabilities of fraud detection systems.

# DATASET AND PREPROCESSING STEPS

The original dataset contained transactions made by credit cards in **September 2013** by European cardholders. The dataset presented transactions that occurred in two days, where we had **492 frauds** out of **284,807 transactions**. The dataset was **highly unbalanced**, the **positive class (frauds)** accounted for **0.172%** of all transactions.

It contained only numerical input variables which were the results of a **PCA transformation**.

Due to confidentiality issues, we were not provided the original features and more background information about the data.

* Features **V1**, **V2**, ... **V28** were the **principal components** obtained with **PCA**.
* The only features which had not been transformed with PCA were **Time** and **Amount**. Feature **Time** contained the seconds elapsed between each transaction and the first transaction in the dataset. The feature **Amount** was the transaction Amount, this feature could be used for example-dependant cost-sensitive learning.
* Feature **Class** was the response variable and it took value **1** in case of fraud and **0** otherwise.

The dataset was then converted based on minute-wise operations in order to create and ensure a traditional time-series dataset.

The new and transformed dataset now contains 33 columns and 2,880 rows. Here are some key details:

* **Time\_Timedelta**: This is the time elapsed in "days, hours, minutes" format.
* **V1 to V28**: These columns represent various features, likely related to transactions or time series data.
* **Total\_Amount, Avg\_Amount, Transaction\_Count, and Fraud\_Count**: These columns contain summary statistics about transactions, including amounts and counts of both total transactions and fraudulent ones.

Finally, the missing values were filled using the **Forward/Backward filling** method in order to ensure continuity without drastic changes in the Time-Series.

# PRELIMINARY ANALYSIS RESULTS

* Both the **ADF (Augmented Dickey-Fuller)** and **KPSS (Kwiatkowski-Phillips-Schmidt-Shin)** tests were conducted to confirm the stationarity of the data. The features which were not strict stationary were made so, through differencing. The stationarity of the complete dataset was then, once again checked using the mentioned statistical tests.

* **ACF and PACF plots**: The autocorrelation and partial autocorrelation plots were generated for the transformed dataset, providing insights into the lag relationships and potential seasonal patterns in the time series data.

* **Seasonal Decomposition**: Each column was decomposed into its seasonal, trend, and residual components. This decomposition helped identify underlying patterns such as cyclic behaviour or trends in both fraud and non-fraud transactions.

* The **seasonality** of fraudulent transactions appears less pronounced, while non-fraudulent transactions exhibit periodic patterns.

* The **ACF and PACF plots** suggest some autoregressive and moving average components, which will be useful in building the ARIMA model.
* Since the dataset is highly imbalanced, SMOTE (Synthetic Minority Oversampling Technique) has been used to balance the data.

Class distribution before SMOTE:

Class

0 227451

1 394

Class distribution after SMOTE:

Class

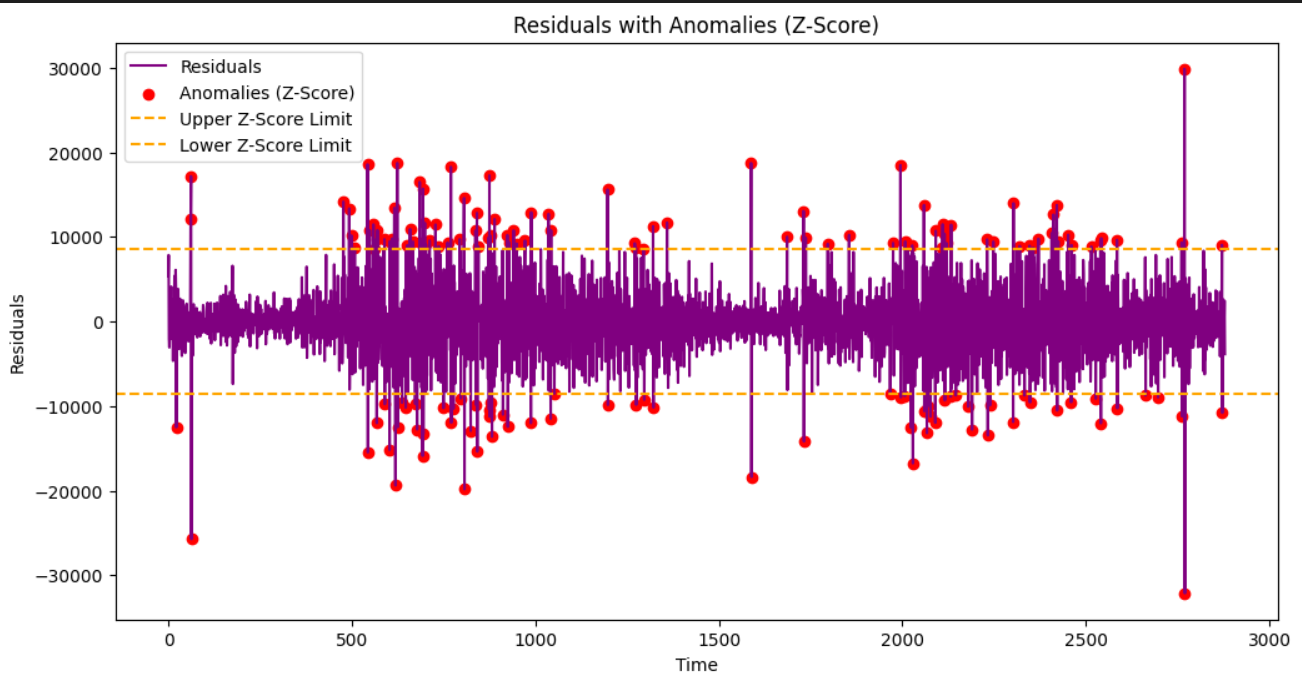
0 227451

1 227451

# Modelling

**ARIMA MODEL**

ARIMA Model is used with hyperparameters of p,d,q=1,01 based acf plots and pacf plots.

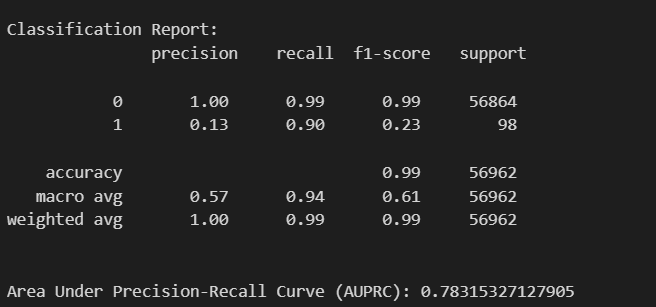


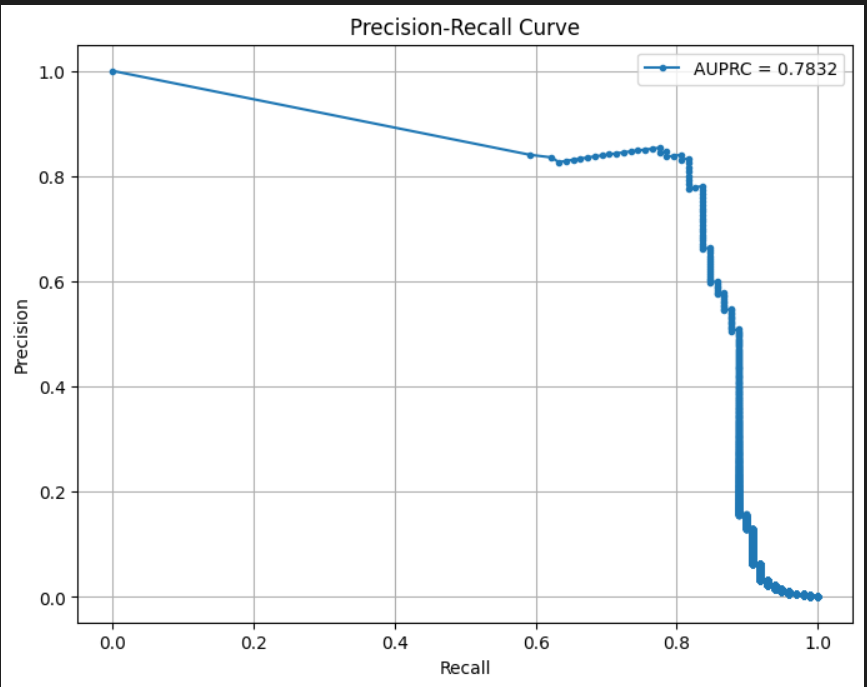
Z-Score is used as threshold to indentify anomalies that point to the timestamp with high potential for fraudulent transactions

These residual anomalies could be used as an additional feature in the stacked ensemble or as a pre-filtering step to reduce noise in the dataset.

Now multiple models are used for traditional classification model. The approach is to use the residual anomalies to increase the performance of the models.

**Logistic Regression**



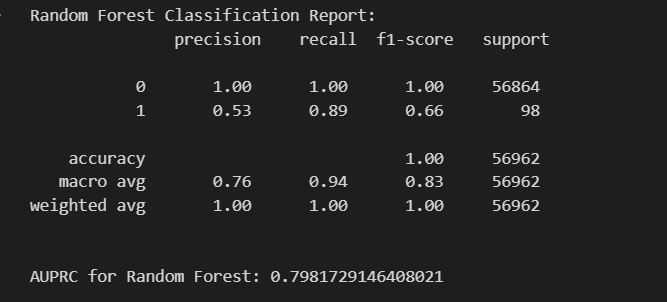


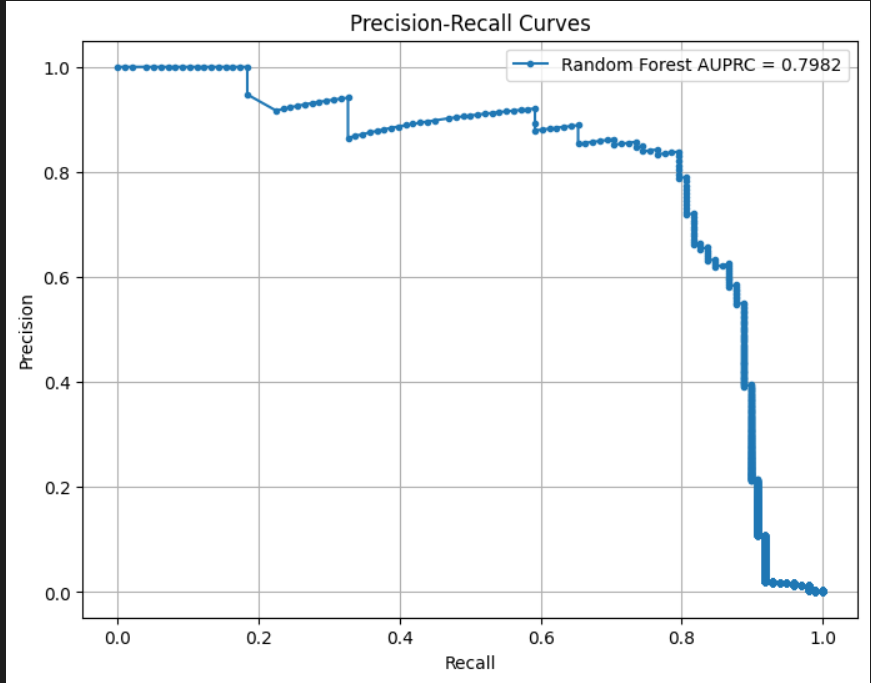
1.) the model provides high accuracy and recall.

2.) the precision and f1 score is very all

3.) this is due to the fact that the model has a high false-positive rate. this means that many non- fradulent transactions are being flagged incorrectly.

**RandomForestClassifier**

****

****

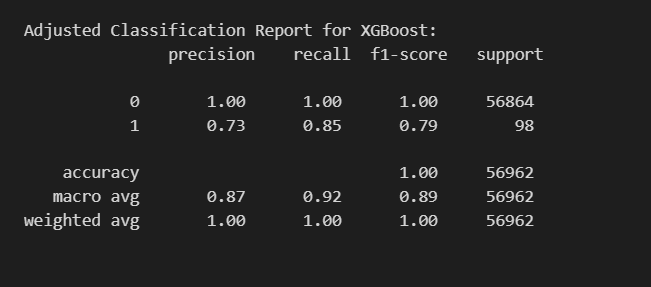
random forest model evaluation insights -

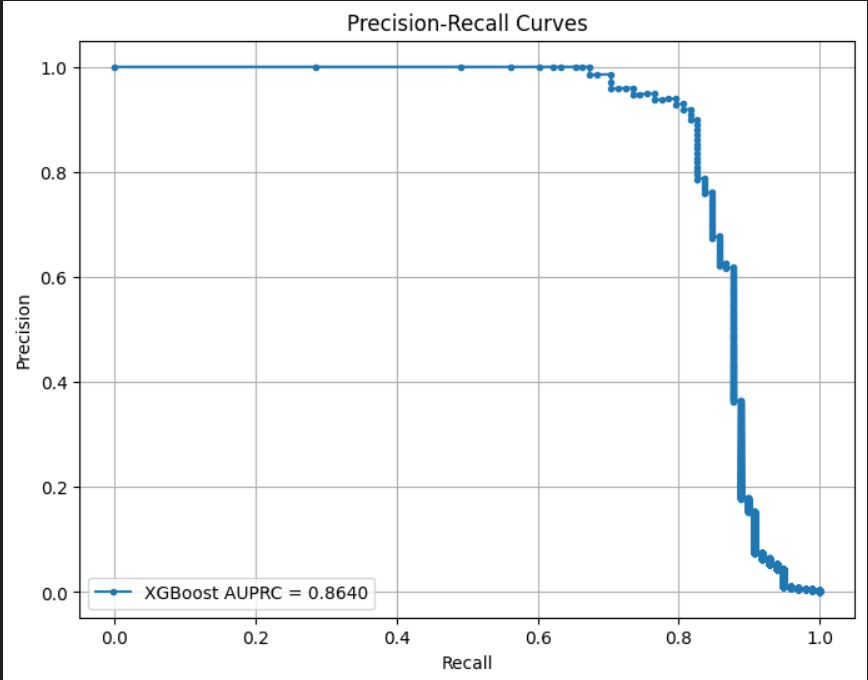
1.) the model is better at handling false positives

2.) the area under the curve has increased as well

however, it is still not upto our standards**.**

**XGBClassifier**

****

****

1.) much better at evaluating false positives

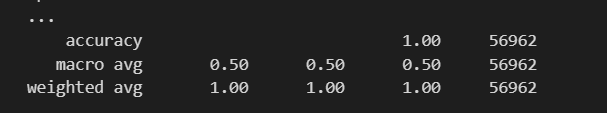
2.) greatly increased the auprc

3.) the recall and f1 score are also decent

using threshold greatly increases the precision and f1score of fradulent cases at slight expense of recall

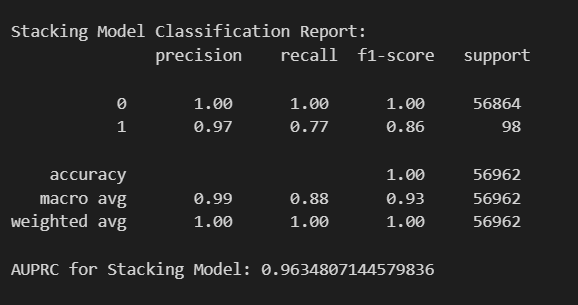
**Deep Learning (FNN and RNN)**

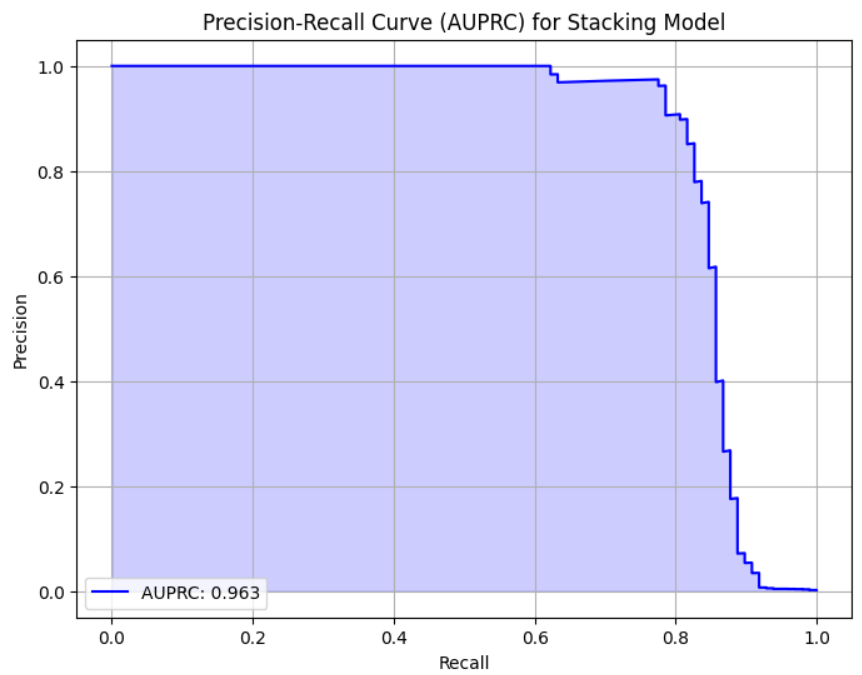
Both FNN and RNN failed to detect Fraudulent transaction likely due to high data imbalance. Even using SMOTE didn’t help the cause

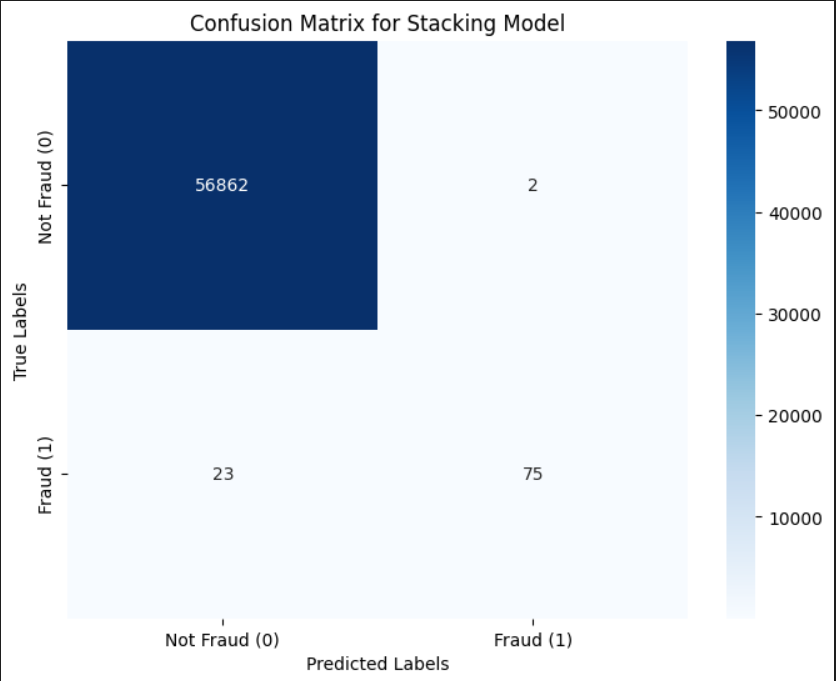


**Model Stacking**

Stacking models with feature engineering allow for a better model that handles the data imbalance well while provide high critical accuracy.







**Conclusion**

The integration of ARIMA residual analysis with a stacked ensemble classification model creates an effective, scalable, and robust fraud detection system due to the complementarity between these two approaches. While ARIMA is excellent at identifying anomalies in time-series data, capturing historical deviations, the stacked ensemble combines the predictive power of multiple classifiers for handling imbalanced datasets to ensure high precision and recall. This again strengthens the detection capabilities of the ensemble approach with ARIMA residuals. In addition to an increased efficiency in terms of accuracy, comprehensiveness is improved so that it allows a cascading effect in fraud detection - having the ARIMA perform first to filter out obvious spam and the ensemble be made to concentrate on critical fraud. Together, they produce a balanced pipeline that can manage larger datasets and generate relevant insights for real-world applications in fraud detection.