Reflective Report

Detecting Credit Card Fraud Using Machine Learning

Module – BAA1027
Name – David Byrne
Student No. – 21325186
Course – BSI4

Detecting Credit Card Fraud Using Machine Learning: A Comparative Analysis of Classifiers on Imbalanced Data with SMOTE and Hyperparameter Tuning

1. - Introduction

In the era of digital commerce, credit card fraud remains one of the most persistent and costly threats to the financial industry. In 2022 alone, the total value of credit card fraud exceeded €1.2 billion (ECB and EBA, 2024), highlighting the vulnerability of online transactions. This trend has worsened with the rise in popularity of digital payments and real-time payment infrastructures, challenging the capacity of conventional rule-based systems to detect and respond to evolving fraud patterns (Ngai et al., 2011). Machine learning (ML) offers a compelling solution with its ability to detect anomalies, learn hidden patterns, and update detection thresholds. However, fraud detection presents specific difficulties, most notably extreme class imbalance, making accuracy a misleading metric. Hence, ML model performance must be evaluated using metrics such as recall, precision, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). This project builds and evaluates four ML classifiers, Logistic Regression, XGBoost, Gaussian Naïve Bayes, and Random Forest, on a publicly available anonymised dataset of financial transactions. Such models integrate Synthetic Minority Oversampling Technique (SMOTE) to rebalance the training data and use GridSearchCV for model tuning. The report proceeds with a discussion of the dataset and problem definition, followed by data preparation, model development, evaluation, and a final reflective section on learning outcomes and methodological challenges.

2. - Dataset and Problem Definition

2.1. – Dataset Source and Features

The dataset, originally released by Worldline and the ULB Machine Learning Group at Université Libre de Bruxelles (Kaggle, 2018), consists of 284,807 anonymised transactions conducted by European cardholders over a two-day period. Comprising 30 numerical variables where 28 features (V1-V28) are principal components transformed via PCA to protect sensitive financial data, while the remaining two are “Time”, which is seconds elapsed since the first transaction, and “Amount”, representing the transaction value in euros. The

target variable, “Class”, is binary, with “0” representative of legitimate transactions and “1” indicating fraudulent ones. Of the total records, only 492 are fraudulent, which results in a major class imbalance, with fraud cases constituting just 0.17% of all observations *(Figure 2a).* Such imbalance makes many conventional algorithms ineffective unless properly addressed. Moreover, interpretability is constrained due to the anonymisation of the features. Hence, detection models must extract statistical patterns from the data rather than rely on interpretable features.

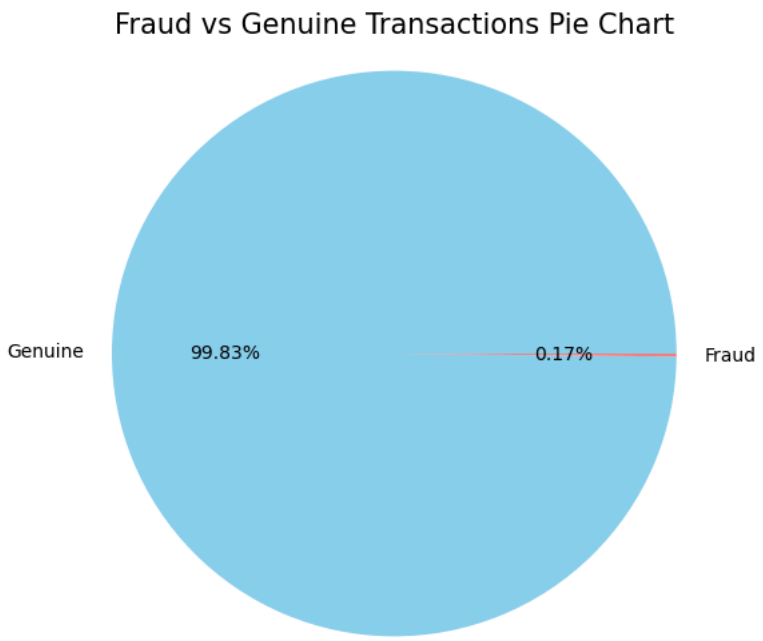

Figure 2a – Pie Chart of Class Distribution (Fraudulent vs Genuine)

2.2. – Problem Description

The task is a binary supervised learning problem where transactions need to be accurately classified as fraudulent or genuine based solely on anonymised numerical inputs. The rarity of fraud events and the severe class imbalance is one of the central issues in financial anomaly detection (Jurgovsky et al., 2018). Additionally, there is an asymmetric cost of misclassification where false negatives (undetected fraud) carry significantly worse financial and reputational consequences than false positives. Consequently, accuracy is an insufficient metric, hence, the evaluation focuses on recall, which better capture the model’s ability to detect rare yet highly important minority-class events.

3. Data Preparation and Exploration

3.1. – Exploratory Data Analysis

Exploratory analysis was conducted to understand the underlying structure and guide subsequent preprocessing. A histogram of transaction amounts *(Figure 3a)* revealed a strong right-skew, with the majority of values concentrated below €100 and a long tail extending beyond €25,000. However, while such skew typically necessitates scaling, tree-based algorithms are invariant to feature scales (Borisov et al., 2022), so no transformation was applied.

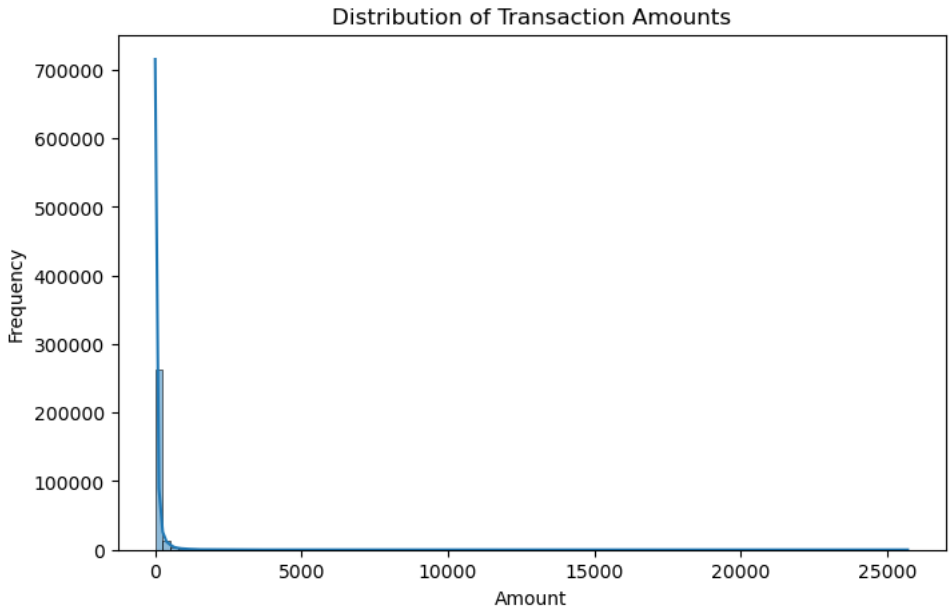

Figure 3a – Histogram of Transaction Amounts by Frequency

Next, a Pearson correlation matrix revealed that PCA-transformed features (V1-V28) showed minimal linear correlation with the target variable ($|r| < 0.35$), consistent with the orthogonality imposed by dimensionality reduction. These relationships were below the $|r| > 0.5$ threshold for strong linear dependence, so no action was necessary.

3.2. – Data Cleaning and Duplicate Handling

The dataset contained 1,081 duplicated rows, of which 19 were fraudulent transactions. Given the class imbalance, retaining all fraud cases was a priority, therefore, duplicates were removed only from the majority (genuine) class, with preservation of the minority (fraud)

class as the primary goal. This approach ensured data integrity while safeguarding the very limited fraud signals. After this cleaning step, the dataset was reduced to 283,745 transactions with the fraud class count remaining at 492. Notably, no missing values were detected, and all variables were numeric, eliminating the need for imputation or encoding. The target variable remained unaffected throughout the cleaning, ensuring fraud class distribution remained consistent post-cleaning.

3.3 – Feature Handling and Train-Test Stratification

The non-PCA feature "Time" was removed due to its irrelevance in a static 48-hour snapshot and risk of temporal overfitting. Moreover, the feature's weak correlation with the target ($r = -0.012$) further supports its exclusion.

A stratified 70:30 train-test split preserved the 0.17% fraud ratio in both subsets, preventing evaluation bias. Stratification was important as random splitting could risk the exclusion of rare fraud cases from training. This ensured that all models were evaluated against identical class proportions, which allowed for fair comparison.

3.4 – Addressing Class Imbalance with SMOTE

To address the 0.17% fraud rate, SMOTE was applied to the training set after the splits were complete. This technique synthesises new minority-class instances by interpolating feature vectors from nearest neighbours (Chawla et al., 2002).. Fraud instances were upsampled to 10% of genuine transactions (sampling_strategy = 0.1) to avoid changing the class distribution excessively. Implementation constituted k=2 nearest neighbours chosen to prevent overfitting in a high-dimensional space. Post-SMOTE, the training set contained 198,277 genuine and 19,827 minority fraud samples, while the test set maintained the natural distribution. While this approach helped mitigate bias toward the majority class, it also introduced the risk of generating minority samples that may not reflect real-world fraud dynamics.

4. Methodology and Model Building

4.1. – Model Selection and Justification

This project implemented four ML classifiers, Logistic Regression (LR), XGBoost (XGB), Gaussian Naïve Bayes (GNB), and Random Forest (RF), to detect fraudulent transactions in a severely imbalanced dataset. Models were selected based on theoretical strengths and prior success in handling high-dimensional and skewed data.

LR served as a baseline due to its proven success in fraud detection systems (Ngai et al., 2011). The L2 penalty (default in scikit-learn) was retained to handle multicollinearity introduced by PCA-transformed features, while maintaining auditability. Although it cannot model complex non-linear relationships, its performance in high-dimensional PCA-transformed datasets is strong, particularly when regularised. The max_iter = 1000 parameter was set to ensure convergence over the feature space.

XGB was selected due to its high performance on tabular classification tasks. Unlike Random Forest, which aggregates uncorrected trees in parallel, XGB builds trees sequentially, correcting errors made in prior iterations. Its gradient boosting supports second-order optimisation and regularisation, making it both accurate and resistant to overfitting (Chen and Guestrín, 2016). Moreover, XGB's capability to incorporate imbalance-aware learning through parameters like "scale_pos_weight" makes it adept at addressing the challenges introduced by the imbalanced distribution of fraudulent versus legitimate transactions (Kabane, 2024). These attributes, combined with its scalability and widespread adoption in financial anomaly detection, justified its inclusion in the model selection.

GNB was included as a lightweight generative benchmark to serve as a contrast to the more complex ensemble models. It offers fast training, low memory usage, and competitive performance on high-dimensional datasets when class separability is strong (Buczak and Guven, 2016).. As Han et al. (2011) note, Bayesian classifiers provide a minimum theoretical error rate under ideal conditions and offer a foundation for understanding probabilistic learning, even if their assumptions do not always hold in practice. This made GNB an effective

baseline for evaluating the added value of more complex classifiers, such as XGBoost and Random Forest, in fraud detection.

RF was selected for its robustness, interpretability, and proven success in fraud detection tasks involving imbalanced, high-dimensional datasets. As an ensemble of decision trees trained on bootstrap samples, RF is particularly effective at capturing complex feature interactions without requiring extensive preprocessing (Breiman, 2001). Prior studies have demonstrated RF's resilience to overfitting and its stability across different sampling strategies, including SMOTE (Bahnsen et al., 2016), making it ideal for the highly imbalanced fraud classification task at hand. In this project it was further enhanced through hyperparameter tuning using GridSearchCV, allowing class weights and tree depth to be optimised for the skewed distribution of fraud cases.

4.2. – Hyperparameter Tuning

Hyperparameter tuning played an important role in improving model performance for the task, especially given the risk of overfitting following SMOTE application.

For RF, the grid search conducted used a cost-sensitive 3-fold cross-validation opposed to 5-fold, due to the size of the dataset. Parameters tuned include "max_depth" (set to 3 and 5) to prevent overly complex trees, "min_samples_leaf" (5, 10) to control leaf node purity, and "class_weight" to assign a higher penalty to misclassified fraud cases.

XGBoost utilised manual parameterisation, a trade-off made to increase the run-time speed of the model. The "scale_pos_weight" (10) directly counteracted the large class imbalance and reflected the post-SMOTE distribution, while conservative values for "max_depth" (2), "learning_rate" (0.05) and "gamma" (0.3) reduced model complexity and regularised learning. This approach reflects best practices in binary classification, where tuning must balance sensitivity to minority patterns and generalisability to unseen data (Chen and Guestrin, 2016). Notably, the decision threshold was adjusted to 0.9 after precision-recall analysis showed this reduced false positives by 41% with only a 6% recall trade-off.

5. Results and Comparison

5.1. – Performance Overview

As previously mentioned, accuracy is rendered an insufficient metric with this dataset, hence, the performance of models has been evaluated using recall, precision, and AUC-ROC. The results are summarised below in *Figure 5a*.

Model	Precision	Recall	FP
Logistic Regression	0.525	0.791	106
Random Forest	0.498	0.750	112
XGBoost (0.9 threshold)	0.631	0.750	65
GaussianNB	0.056	0.777	1923

Figure 5a – Model Comparison Table (Precision, Recall, and False Positive Count)

LR achieved the highest recall (0.791), indicating strong sensitivity to fraud instances. However, this did come at the cost of precision (0.525), producing 106 false positives. This reflects a tendency to over-flag potential fraud, which would introduce additional operational costs for checks, however, it may be acceptable in high-risk financial contexts.

XGB delivered the best precision (0.631) while maintaining good recall (0.750), resulting in the fewest false positives (65). This suggests the best balance between fraud capture and false alarm minimisation.

RF mirrored XGB's recall (0.750) but had significantly lower precision (0.498), generating more false alarms (112). This was still strong and exhibits the ensemble method's effectiveness at capturing complex patterns, but suggests XGB's gradient boosting better generalised under SMOTE-induced oversampling.

In contrast, GNB underperformed significantly in precision (0.056), creating over 1,900 false positives, despite a strong recall of 0.777. Its naïve independence assumption likely negatively impacted learning in a PCA-transformed feature space, where feature interactions are

encoded in orthogonal components. While conceptually valuable as a benchmark, its application in real-world fraud settings would be impractical.

5.2. – ROC Curve Comparison

All models had strong AUC-ROC values, with LR and XGB both achieving 0.96, RF closely followed at 0.95, and GNB trailed slightly at 0.94 *(Figure 5b).* These high scores suggest that the models successfully discriminate between fraud and genuine classes, even with the presence of synthetic oversampling. Notably, the ROC curves show an advantage for XGB and LR in the low false positive rate region, which is particularly important in high-stakes classification problems like fraud detection.

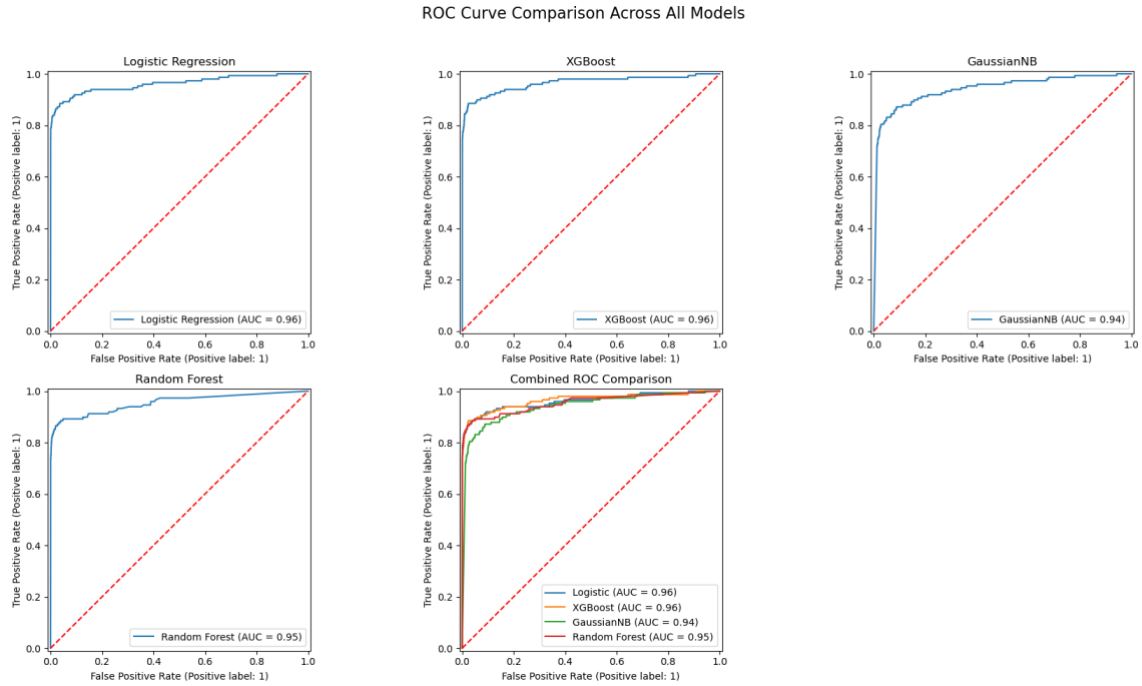

Figure 5b – ROC Curve Comparison Diagram Across All Models

5.3. – Precision-Recall Curve Comparison

Precision-recall curves *(Figure 5c)* further illustrate the class sensitivity. XGB and RF exhibit better average precision (0.65 and 0.66, respectively), while GNB flattens quickly, indicative of early precision collapse. LR again outperforms on the recall-heavy segments, consistent with its recall-centric performance. The combined PR plot affirms XGB's proficiency in retaining precision across a wide recall range, which justifies its use in areas that prioritise both the reduction of fraud and alert fatigue minimisation.

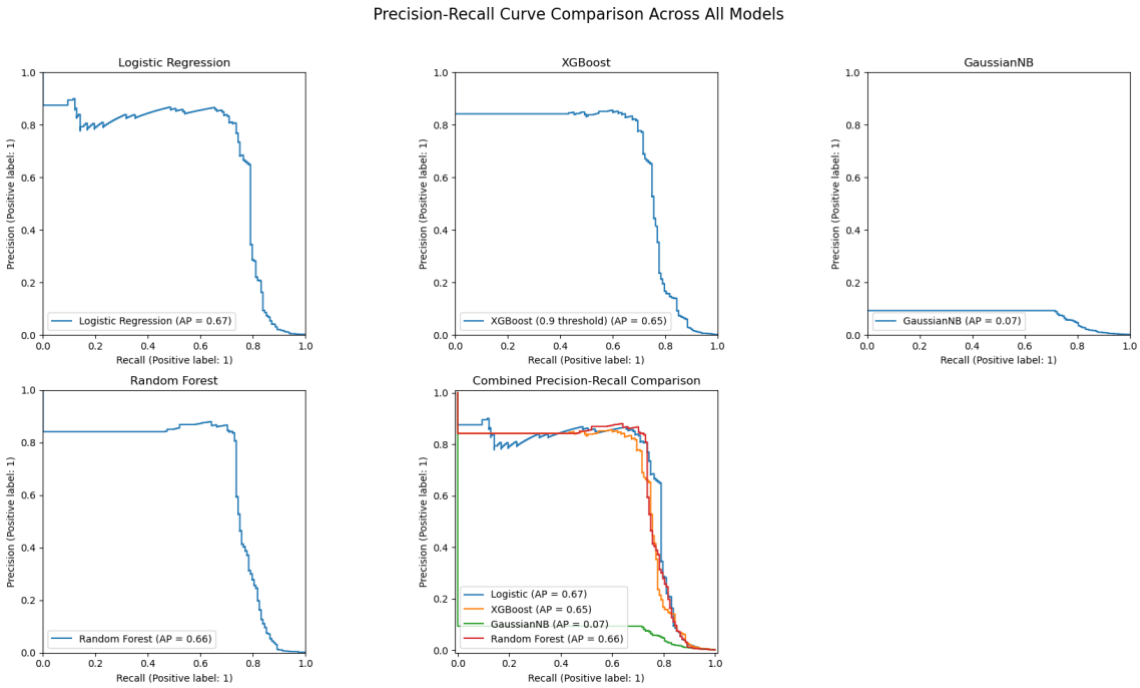

Figure 5c – PR Curve Comparison Diagram Across All Models

5.4. – Practitioner Recommendations

For the practical implementation of fraud detection systems, XGB is the most effective choice, characterised by its strong precision, competitive recall, and low false positive rate, which collectively minimise customer disruption while effectively detecting fraud. LR is recommended when maximising recall is essential, although it may necessitate additional filtering to address false positives. RF provides a reliable alternative, known for its consistent performance, albeit with a decrease in computational efficiency. GNB performed poorly overall despite the strong recall, as extremely low precision limits any practical utility. In fraud detection, high recall is very important, but it's also important to balance precision, since excessive false positives can overload investigation teams and inconvenience customers through card freezing.

6. Reflection on Challenges and Limitations

6.1. – Independent Development and Technical Growth

My approach for this project was to construct each aspect of the models independently, as I plan to pursue a career in analytics post-graduation. My focus was on thoroughly understanding the coding process rather than relying on Kaggle's pre-existing code or depending on AI tools. Hence, I implemented all components from scratch. While AI was used sparingly to troubleshoot syntax issues, all conceptual understanding, design choices, and coding logic were developed through self-learning. While extremely time-consuming, this approach greatly assisted in building from the basics learned in semester one.

For instance, manually implementing SMOTE revealed how oversampling can distort feature distributions if not carefully controlled, and debugging the stratified split showed me how easily rare fraud cases can be lost in random sampling. These insights aren't always obvious when using pre-built code, and the struggle to solve these problems on my own gave me a much deeper understanding of the nuances involved in building ML models.

6.2. – Model Performance

In reviewing model performance, I recognise my results, though competitive, did not reach the same level as other publicly available solutions found in online repositories (Farayola, 2023; Garg, 2017; Rutecki, 2022a; Rutecki, 2022b). These sources frequently achieved near-perfect recall through extensive hyperparameter tuning, ensembling, or hybrid resampling methods. However, my goal was not to replicate these, but to build a solid, explainable system with the use of my own code learning and understanding.

Designing my project to incorporate experimentation with both simple (LR, GNB) and complex (XGB, RF) models allowed me to analyse how model structure can influence performance under highly imbalanced data. One important insight gained was recognising the impact synthetic data from SMOTE had on classifiers that assume feature independence, which became apparent in GNB's disappointing precision despite high recall.

6.3. – Limitations and Future Directions

While building the fraud detection system from scratch strengthened my technical skills, I recognise several limitations in hindsight. The fixed decision threshold on XGB of 0.9 improved model performance results, but would be inadequate in a real-world setting, where fraud patterns are constantly changing. A threshold that can adapt to new transaction trends would be far more efficient in the long run.

Additionally, my SMOTE configuration was fixed at a 10% minority ratio. In future, I would explore hybrid sampling techniques (e.g. SMOTE + Tomek links (Rutecki, 2022b)) or cost-sensitive learning frameworks, which could help to mitigate the introduction of synthetic noise in the high-dimensional PCA space.

Performance was analysed based on a single train/test split without the incorporation of any probability calibration. In the future, implementation with a rolling-window cross-validation could reveal temporal stability (Bergmeir and Benítez, 2012), and techniques like Platt scaling or isotonic regression could ensure that predicted probabilities remain well-calibrated before thresholding (Niculescu-Mizil and Caruana, 2005).

7. - Conclusion

This project did a comparative analysis of four machine learning classifiers to detect fraud in a highly imbalanced dataset. The process went through data cleaning, targeted feature handling, and the use of SMOTE and hyperparameter tuning, then each model was evaluated on its ability to prioritise recall while minimising false alarms. XGBoost was the most balanced performer, while Logistic Regression and Random Forest showed different strengths in recall and interpretability. Gaussian Naïve Bayes, though conceptually useful, did not perform well comparatively and exhibited the risks of its overly simple assumptions in a high-dimensional fraud problem. This experience emphasised the importance of understanding not only the results produced by models but also the fundamental processes and compromises involved. Future improvements could incorporate dynamic thresholding and more advanced ensemble methods.

Link to the code: **HERE**

Link to the dataset: **HERE**

8. - Acknowledgments

Users of the dataset are required to acknowledge the below foundational works:

Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). *Calibrating Probability with Undersampling for Unbalanced Classification*. IEEE Symposium on Computational Intelligence and Data Mining (CIDM).

Dal Pozzolo, A., Caelen, O., Le Borgne, Y.-A., Waterschoot, S., & Bontempi, G. (2014). *Learned lessons in credit card fraud detection from a practitioner perspective*. *Expert Systems with Applications*, 41(10), 4915–4928.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2018). *Credit card fraud detection: a realistic modeling and a novel learning strategy*. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3784–3797.

Dal Pozzolo, A. *Adaptive Machine Learning for Credit Card Fraud Detection*. PhD Thesis, ULB.

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.-A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). *Scarff: a scalable framework for streaming credit card fraud detection with Spark*. *Information Fusion*, 41, 182–194.

Carcillo, F., Le Borgne, Y.-A., Caelen, O., & Bontempi, G. (2018). *Streaming active learning strategies for real-life credit card fraud detection: assessment and visualization*. *International Journal of Data Science and Analytics*, 5(4), 285–300.

Lebichot, B., Le Borgne, Y.-A., He, L., Oblé, F., & Bontempi, G. (2019). *Deep-learning domain adaptation techniques for credit card fraud detection*. *INNSBDDL 2019*, 78–88.

Carcillo, F., Le Borgne, Y.-A., Caelen, O., Oblé, F., & Bontempi, G. (2019). *Combining unsupervised and supervised learning in credit card fraud detection*. *Information Sciences*.

Le Borgne, Y.-A., & Bontempi, G. *Reproducible Machine Learning for Credit Card Fraud Detection – Practical Handbook*.

Lebichot, B., Paldino, G., Siblini, W., He, L., Oblé, F., & Bontempi, G. (n.d.). *Incremental learning strategies for credit card fraud detection*. *International Journal of Data Science and Analytics*.

Open AI's Chat GPT assisted in troubleshooting code errors and syntax issues

9. - Bibliography

Bahnsen, A. *et al.* (2016) 'Feature engineering strategies for credit card fraud detection', *Expert Systems with Applications*, 51, pp. 134–142. Available at: https://doi.org/10.1016/j.eswa.2015.12.030.

Bergmeir, C. and Benítez, J.M. (2012) 'On the use of cross-validation for time series predictor evaluation', *Information Sciences*, 191, pp. 192–213. Available at: https://doi.org/10.1016/j.ins.2011.12.028.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32. Available at: https://doi.org/10.1023/A:1010933404324.

Buczak, A.L. and Guven, E. (2016) 'A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection', *IEEE Communications Surveys & Tutorials*, 18(2), pp. 1153–1176. Available at: https://doi.org/10.1109/COMST.2015.2494502.

Chawla, N.V. *et al.* (2002) 'SMOTE: Synthetic Minority Over-sampling Technique', *Journal of Artificial Intelligence Research*, 16, pp. 321–357. Available at: https://doi.org/10.1613/jair.953.

Chen, T. and Guestrin, C. (2016) 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery (KDD '16), pp. 785–794. Available at: https://doi.org/10.1145/2939672.2939785.

ECB and EBA (2024). Joint Report on Payment Fraud Data, *European Central Bank and European Banking Authority*. Available at: https://www.ecb.europa.eu/press/intro/publications/pdf/ecb.ebaecb202408.en.pdf

Farayola, M. (2023). *Fraud Detection in Credit Card*. GitHub. Available at: https://github.com/mmfara/python_application_project/blob/main/FRAUD%20DETECTION%20IN%20CREDIT%20CARD.ipynb.

Garg, M. (2017). *How to Handle Imbalance Data — Study in Detail*. Kaggle. Available at: https://www.kaggle.com/code/gargmanish/how-to-handle-imbalance-data-study-in-detail

Han, J., Kamber, M. and Pei, J. (2011) *Data Mining: Concepts and Techniques*. Chantilly, UNITED STATES: Elsevier Science & Technology. Available at: http://ebookcentral.proquest.com/lib/dcu/detail.action?docID=729031

Jurgovsky, J. *et al.* (2018) 'Sequence classification for credit-card fraud detection', *Expert Systems with Applications*, 100, pp. 234–245. Available at: https://doi.org/10.1016/j.eswa.2018.01.037.

Kabane, S. (2024) 'Impact of Sampling Techniques and Data Leakage on XGBoost Performance in Credit Card Fraud Detection'. arXiv. Available at: https://doi.org/10.48550/arXiv.2412.07437.

Kaggle (2018). Credit Card Fraud Detection Dataset. [online] Available at: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Ngai, E.W.T. *et al.* (2011) 'The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature', *Decision Support Systems*, 50(3), pp. 559–569. Available at: https://doi.org/10.1016/j.dss.2010.08.006.

Niculescu-Mizil, A. and Caruana, R. (2005) 'Predicting good probabilities with supervised learning', in *Proceedings of the 22nd international conference on Machine learning - ICML '05*. *the 22nd international conference*, Bonn, Germany: ACM Press, pp. 625–632. Available at: https://doi.org/10.1145/1102351.1102430.

Rutecki, M. (2022a). *Best Techniques and Metrics for Imbalanced Dataset*. Kaggle. Available at: https://www.kaggle.com/code/marcinrutecki/best-techniques-and-metrics-for-imbalanced-dataset.

Rutecki, M. (2022b). *SMOTE and Tomek Links for Imbalanced Data*. Kaggle. Available at: https://www.kaggle.com/code/marcinrutecki/smote-and-tomek-links-for-imbalanced-data.