National College of Ireland

Cost – Sensitive Hyperparameter Tuning for XGBoost in Credit Card Fraud Detection

MSc Research Project
MSc in AI for Business

Thi Thuy Hang Nguyen
Student ID: 23347325

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Thi Thuy Hang Nguyen

Student ID: 23347325

Programme: MSCAIBUS1 **Year:** 2024-2025

Module: Practicum

Supervisor: Rejwanul Haque

Submission Due Date: 11th August 2025

Project Title: Cost-Sensitive Hyperparameter Tuning for XGBoost in Credit Card Fraud Detection

Word Count: ……**6515**…… **Page Count**……20……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Thi Thuy Hang Nguyen

Date: 12/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only

Signature:	Thi Thuy Hang Nguyen
Date:	12/09/2025
Penalty Applied (if applicable):	

Cost – Sensitive Hyperparameter Tuning for XGBoost in Credit Card Fraud Detection

Thi Thuy Hang Nguyen
23347325

Abstract

Credit card fraud detection is a daunting task considering the highly imbalanced nature of transaction data and the cost of false negatives being prohibitive. Traditional machine learning classifiers fail to perform well in such a scenario, leading to poor minority class performance. To overcome this, the present research work demonstrates an enhanced fraud detection system with Extreme Gradient Boosting (XGBoost) using cost-sensitive learning and SMOTE (Synthetic Minority Oversampling Technique). A baseline XGBoost classifier is first developed without class weighting. A cost-sensitive classifier is then developed using the scale_pos_weight parameter to increase the sensitivity of the minority class. SMOTE is then applied to further balance the training data to reduce false negatives. RandomizedSearchCV is employed for hyperparameter tuning. Classifiers are evaluated on precision, recall, F1-score, ROC-AUC, and the confusion matrix. Results indicate that SMOTE with cost-sensitive XGBoost significantly improves recall and AUC in identifying fraudulent transactions and is thus highly suitable for application in real-world financial fraud detection systems.

Keywords: credit card fraud detection, XGBoost, cost-sensitive learning, SMOTE, hyperparameter tuning, imbalanced data.

1 Introduction

Fraud is a global problem that targets every corner of the globe and every business sector [1]. With the exponential growth of digital banking, online business, and electronic payments, new avenues have emerged for cyber offenders to exploit weaknesses in financial systems [2]. Modern fraud schemes are ever more sophisticated, targeting individuals, businesses, and government agencies [3]. Identity theft, phishing, embezzlement, and money laundering are

some of the most common forms of fraud. The rapid evolution of these schemes presents tremendous challenges to traditional fraud detection systems, which often lag the ingenuity of fraudsters today.

Online payment fraud is estimated to cost as much as $260 billion worldwide by 2025, and financial institutions and merchants are projected to lose over $343 billion between 2023 and 2027 [4]. Numbers like these highlight the necessity for advanced fraud prevention. Traditional detection techniques are no longer as successful due to the growing complexity and volume of fraud [5].

Artificial intelligence (AI) offers a promising solution. Over the past decade, AI technologies such as machine learning (ML), computer vision, and natural language processing have become increasingly integrated into sectors like healthcare, logistics, education, and government. In the financial industry, AI has proven valuable for processing vast datasets and uncovering complex behavioral patterns. AI-driven tools enhance customer service through features like chatbots and personalized recommendations, while also playing a vital role in managing risk and enhancing security. A key application of AI in finance is fraud detection, where systems analyze large-scale transaction data in real time to flag suspicious activity and issue alerts [6]. ML algorithms can identify fraud by learning from historical transaction data, thereby helping banks reduce losses and protect sensitive customer information.

Despite growing adoption, there is still limited research into the performance of different machine learning models for detecting fraudulent banking transactions. The current study tries to address that gap with the optimization of XGBoost for credit card fraud detection through cost-sensitive hyperparameter tuning. Instead of striving for the highest overall accuracy, the approach aims to minimize the financial cost of undetected fraud. By tuning model parameters at the cost level, this study strives to enhance fraud detection and help banks make more informed and cost-saving decisions.

Research Questions are:

(1) How does cost-sensitive hyperparameter tuning improve XGBoost's performance in detecting credit card fraud compared to default parameters?

(2) Which hyperparameters have the most significant impact on reducing false negatives (missed fraud cases) in XGBoost?

(3) How can tuning the parameters in a fraud detection model help businesses balance between protecting assets and maintaining customer experience?

Section 2 provides a synthesis and analysis of prior research on machine learning in the banking sector for fraud detection. **Section 3** begins by providing an overview of the research methodology. **Section 4** provides implementation for the project. **Section 5** assesses the efficacy of these machine learning algorithms in fraud detection. **Section 6** evaluates and compares the results of the models. **Section 7** provides the research findings and provides prospective research to enhances AI fraud prevention inside the banking sector.

2 Related Work

2.1 Overview of Fraud Detection in Financial Transactions.

Fraud detection systems (FDSs) are essential, as they protect financial integrity by detecting and reporting fraud. Audit procedures have changed from manual to automated; however, they are still not perfect. The manual functions are exemplified by discovery sampling, which was very time-consuming and tiresome. To improve efficiency, however, automated FDSs have been created. The very first automated systems were constrained by the fact that they were rule-based (7).

Modern FDSs employ data mining techniques like classification (neural networks, support vector machines) and clustering (K-means) as well as statistical, machine learning, and high-performance computer technologies for implementation. The combination of anomaly-based and misuse-based can also give detection with higher accuracy. Most of the recent literature focuses on a hybrid model which employs misuse detection supplemented with anomaly detection to cater for the faults of each user model.

The core of fraud discovery in electronic fund transfers is the use of high-tech analytics, machine learning models, and AI for detecting irregular patterns and behavior. By processing massive quantities of transaction data at lightning speed, the financial system can quickly uncover and isolate fraud, thus helping it protect assets, maintain trust, and uphold financial integrity [8].

Fraud detection in financial transactions primarily focuses on identifying and preventing unauthorized transactions to ensure security. The implementation of this specification will require continuous checkups for any abnormalities based on the current technology developed such as machine learning. The most typical kinds of financial fraud are credit card fraud, identity theft, phishing, account takeovers, money laundering, and insider fraud [9]. The list of current most effective detection techniques includes, among others, data analysis, machine learning algorithms, real-time monitoring, and customer profiling. The best fraud prevention practices list notably includes multi-factor authentication. The list also

includes ongoing security updates, employee and client education, audits, and coordination with other financial institutions. These practices are supposed to be implemented by businesses if they want to efficiently detect and prevent fraud.

2.2 Machine Learning in Fraud

To avoid fraudulent transactions and identify credit card fraud, a variety of approaches have been suggested by researchers.

Hashemi et al. considered the utilization of class weight-tuning hyperparameters to manage the relative priority of true over false transactions. The researchers used Bayesian optimization as the method of tuning such hyperparameters in the case of a real-world application. The study contribution proposes weight-tuning as an effective method of preprocessing imbalanced datasets and employs a voting ensemble method that utilizes CatBoost and XGBoost for enhancing the performance of the LightGBM method. The paper also employs deep learning algorithms in conducting hyperparameter search. The findings indicate that the method introduced in this study, which involves deep learning and Bayesian optimization in hyperparameter search, has a significant edge in comparison to the state-of-the-art methods under consideration [10].

Noviandy et al.'s study investigated the use of machine learning for credit card fraud detection using XGBoost and various data augmentation techniques to balance the dataset with fraudulent samples and improve classification. They determined that XGBoost, when coupled with the SMOTE ENN technique, gave the greatest performance. The approach successfully addresses class imbalance by oversampling the minority class through SMOTE and undersampling the majority class through ENN to address possible noise. These findings demonstrate the capability of the solution to significantly improve financial management tasks by equipping finance managers and accountants with predictive models for informed decision-making, resource planning, and pre-emptive fraud management. This study [11] facilitates more effective fraud detection processes, thereby safeguarding financial institutions and companies from fraud.

Zhang et al. addressed the growing issue of detecting more and more complex fraud in a variety of financial transactions, including credit cards, bankers' cheques, and internet fund transfers. The study proposes the application of machine learning algorithms, i.e., K-Nearest Neighbors (KNN), Naïve Bayes, and Support Vector Machines (SVM), for enhancing the

detection of fraudulent transactions. The principal findings indicate that the SVM algorithm performed best with 99.23% accuracy. This work emphasizes the effectiveness of machine learning, i.e., SVM, for the development of efficient fraud detection systems, which is a promising solution for improving financial security [12].

Minastireanu & Elena-Adriana attempted to classify and analyze machine learning algorithms appropriate for online bank fraud detection based on parameters of high accuracy, high coverage, and low cost. They used the systematic quantitative literature review method to compare research papers related to fraud detection that are relevant. The findings revealed that supervised learning methods like Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Decision Trees were accurate and complete but costly. The study also highlighted the hybridization of the most popular machine learning methods for increasing the efficacy of fraud detection in other significant domains and analyzed the same using their own data [13].

Moreira et al. researched potential directions of technological assistance for the analysis of fraudulent transactions using the bank's historical database. Statistical models were used to analyze data and investigate observations in a database containing more than six million financial transactions. Most significant findings showed that Logistic Regression and K-Nearest Neighbors (KNN) models were promising to use in the future at the bank [14].

Thennakoon et al. presented a new credit card fraud detection system that can identify four patterns of fraudulent transactions. The suggested system is an optimum algorithm-based system, overcoming the limitations of existing credit card fraud detection studies. The research used a thorough comparison of machine learning algorithms, based on effective performance measures, to identify the most appropriate algorithms for each of the four fraud patterns. The results revealed a substantial contribution of resampling techniques in attaining relatively greater classifier performance. To our surprise, Support Vector Machines, Naïve Bayes, K-Nearest Neighbors, and Logistic Regression were the four machine learning algorithms with the highest accuracy rates [15].

Gyamfi & Nana Kwame spoke about the types of fraud banks are prone to, looking at the use of data mining tools in early fraud detection using bank data accumulated over time periods. In the research work here, the supervised learning technique Support Vector Machines with Spark (SVM-S) was used in modeling normal and anomalous customer

behavior and then verifying the legitimacy of incoming transactions. The study findings demonstrated the value of these methods in banking fraud prevention in big data environments. Experimental results under the research indicated that SVM-S exhibited superior predictive capability compared to Back Propagation Networks (BPN) [16].

Chen et al. applied stepwise regression for variable selection and then constructed classification models using logistic regression, Support Vector Machines (SVM), and Decision Trees for comparison. The research integrated financial and non-financial variables for developing a predictive model of fraudulent financial statements. The empirical findings revealed that the SVM model yielded the lowest Type I error rate, and the Decision Tree C5.0 model achieved the highest classification performance based on Type II error and overall classification accuracy [17].

Manop Phankokkruad [18] tackled the problem of imbalanced data in the classification of breast cancer by developing a cost–sensitive Extreme Gradient Boosting (XGBoost) model. The improved model incorporates cost–sensitive learning to prefer correct classification of minority classes, which are frequently underrepresented in medical datasets. With four various breast cancer datasets, the research demonstrated that the cost–sensitive XGBoost greatly outperformed the conventional XGBoost, particularly on accuracy, precision, recall, and AUC scores. The efficacy of the model was confirmed using hyperparameter tuning and k – fold cross – validation, highlighting its prospects for enhancing early diagnosis of breast cancer.

Gupta et al. [19] investigated the integration of XGBoost and SMOTE to enhance credit card fraud detection on highly imbalanced datasets. Their study revealed that while XGBoost performed well on its own, the application of SOMTE significantly improved the fraud detection performance of the model by balancing class distribution. Using a real–world dataset, they achieved near–perfect accuracy (99.97%), precision, recall, and F1 scores when they applied SMOTE, which suggests the importance of handling class imbalance in fraud detection. This study adds to the value of combining ensemble learning with data resampling techniques to achieve effective financial anomaly detection.

Leo et al. [25] researched the application of machine learning in banking risk management through a review of the literature. The research reported that machine learning is assuming a more important role in business, with many applications already being utilized

and many others being explored. It also aims to identify gaps in the current body of research where machine learning has not been adequately leveraged, representing opportunities for future research. The article discloses that although machine learning has been used on many types of risks like credit, market, operational, and liquidity risks, the scope of application does not yet match the heightened focus on both risk management and machine learning within the industry. There are many areas involving banking risk that are still unexplored, holding considerable scope for focused machine learning solutions.

Recent works have increasingly focused on using graph neural networks (GNNs) to detect fraud in mobile social networks. However, a significant challenge in this context is the inherent class imbalance, where fraudulent samples are significantly outnumbered by benign instances. Hu et al. [26] address this issue by proposing a cost–sensitive GNN framework that combines reinforcement learning (RL) – based neighbor sampling and cost-sensitive learning to enhance model performance in imbalanced conditions. The RL–based sampler selectively filters neighbors based on learned similarity metrics, reducing the negative effect of majority class neighbors. At the same time, a cost matrix is learned and optimized through backpropagation to reweigh misclassification costs, enhancing minority class detection. Their model outperformed standard GNNs and state-of-the-art imbalanced learning techniques on two real-world telecom fraud datasets. This work shows the effectiveness of cost-sensitive learning in GNNs and suggests promising directions for future research in graph-based fraud detection under class imbalance conditions.

2.3 Research Niche

Credit card fraud detection is still very difficult due to the highly unbalanced class problem in fraud datasets, where illegal transactions are much less frequent than legal ones. Various machine learning methods—such as logistic regression, random forests, and deep learning—have been broadly utilized, but XGBoost has attracted some attention for its robustness and capability of dealing with imbalanced data. In addition, XGBoost is the most efficient algorithm. Nevertheless, only a few studies are dedicated to the adaptation of XGBoost by means of cost-sensitive strategies in fraud detection. Most current research is focused on improving overall performance figures, like AUC-ROC and F1-score. These are good, but still, they usually miss the point of the real financial cost of misclassification, especially the big losses that come from false negatives—where the fraud is not detected. Cost-sensitive learning, which sets different punishments for mistaken parts depending on their influence, could be a solution. On the other hand, there is still a lack of research about

how the integration of XGBoost and the process of hyperparameter tuning can work through the use of cost-sensitive learning. Therefore, in this paper, we are going to address the issue by suggesting a cost-sensitive hyperparameter search framework for XGBoost, which not only improves the predictive power but also results in a better understanding of financial risk that the model shares with the learning process. The suggested method permits a more grounded and efficient response to the issue of fraud concealment as it is able to directly factor in the cost of the fraud missed during the learning process of the model.

3 Research Methodology

3.1 Extreme Gradient Boosting

Fraud detection is one of the most significant challenges tackled by machine learning and data analysis in the finance, banking, and e-commerce industries. The algorithms of machine learning can enable the automation of fraud detection, thus reduce financial loss and increasing the productivity of the business. The authors of this article utilize the XGBoost algorithm to recognize fraud in credit card transactions.

XGBoost is presently one of the most efficient classifiers that have been used in multiple domains, and the tool has also been utilized for fraud detection with success [20]. XGBoost is an acronym for eXtreme Gradient Boosting, and it represents a distributed and open-source machine learning library that employs gradient boosted decision trees, a model that is based on gradient descent, a supervised learning boosting algorithm. It is widely recognized for its speed, effectiveness, and good scalability with big datasets. XGBoost differs from the usual gradient boosting implementation in scikit-learn through several features that include parallel and distributed computing support via in-memory blocks, which allows efficient processing of multiple machines or external storage by using out-of-core computing, and is also integrated with distributed frameworks such as Apache Spark, Dask, and Kubernetes for faster training. Besides that, XGBoost also uses a cache-aware prefetching technique, which greatly decreases the runtime for big datasets, and it is usually far ahead of other frameworks by more than ten times on a single machine, while still being scalable for datasets of the billion-range. The built-in regularization in the learning objective is another important advantage that improves model performance over traditional gradient boosting, together with hyperparameter tuning for additional optimization. On top of it, XGBoost also features a sparsity-aware algorithm that automatically solves the problem of missing values by learning the best default directions, thus it becomes even more robust for

real-world datasets that contain incomplete data. All these features make XGBoost an excellent option for high-performance, scalable machine learning jobs.

XGBoost is a gradient boosting evolution carrying forward the performance further through the utilization of algorithmic improvements and efficient data processing. The first step of this is by dividing the dataset into a training subset and a testing subset and then converting the data into XGBoost's own DMatrix format, which is a thoroughly optimized data structure resulting in memory and speed efficiency. After the data is ready an XGBoost model has been called with a suitable objective function, for example, multi: softmax in case of the multiclass classification or binary: logistic in case of the binary classification [21]. The model fits on the training set, and the test set predictions are generated. The performance is measured by metrics such as accuracy, precision, recall, and F1-score, besides visualization tools like confusion matrices, which help to get a picture of true and false predictions. To improve the model, a hyperparameter tuning step is undertaken. This is done mostly by the grid search together with the cross-validation, which relies on the different parameter combinations systematically explored to reach the best predictive performance.

Choosing the right hyperparameters is very important to get the most out of XGBoost. The learning rate is one of the most important parameters here; it decides how much influence each tree in the ensemble should have. Lower values (e.g., 0.01-0.3) reduce the fit to noise and therefore require more trees, while higher values speed up the learning but may cause errors in the model. The n_estimators parameter decides the number of boosting rounds (trees) in the model; changing this number to a higher value allows the model to be more complex but makes it necessary to use a lower learning rate to avoid overfitting. Also, the gamma parameter indicates the smallest loss reduction that can result from making additional splits in the process of finding the optimum; bigger numbers imply stricter pruning, that is, fewer parts of the tree can be used, which leads to less overfitting, but there is also a chance of underfitting if the number is too high. Furthermore, max_depth controls how deep the trees can go, and deeper trees can capture more complicated patterns, but they also increase the overfitting risk (default is 6). Through various approaches like grid search or randomized search, tuning these hyperparameters could help ensure that the model generalizes well and performs optimally when tested on different datasets.

XGBoost retains its top presence in financial fraud detection as it solves the problem of class imbalance, can work with transaction data with many features, and still gives

understandable results [22]. Regularization parameters of the algorithm, like gamma and max_depth, are used for overfitting control, while the fact that XGBoost supports missing values by default makes it possible to use real-world payment datasets without any hurdles [23]. Performance evaluation experiments indicate XGBoost's superiority over conventional models in fraud tasks, with implementations that are more efficient than random forests [24]. The method's ability to perform parallel processing also makes it possible to carry out real-time fraud detection tasks in monitoring pipelines.

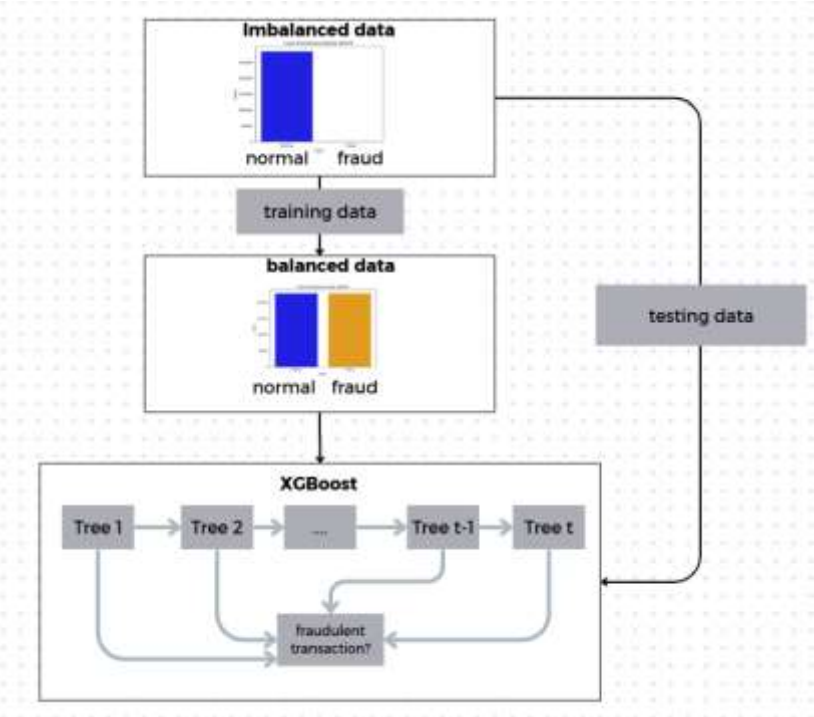

Figure 1. Flowchart of XGBoost for fraud detection

3.2 Cost – sensitive Imbalanced Classification

Cost – sensitive learning is a machine learning method aimed at dealing with situations where errors have different weights in the decision. It is mainly suitable for imbalanced classification problems, where one class (the minority) is represented to a much lesser extent than the majority [18]. This approach is implemented by allocating more weight to errors in the minority class, thus utilizing a cost matrix that depicts the intensity of different errors. The goal is to reduce the total cost of the mistakes rather than only increasing the accuracy.

The costs of false positive (FP), false negative (FN), true positive (TP), and true negative (TN) can be represented in a cost matrix shown in Table 1..

Table 1. An example of a cost matrix for classification

	Actual negative	Actual Positive
Predict negative	TP	FN

Predict positive	FP	TN

4 Implementation

This paper aims to develop the cost-sensitive fraud detection model using the gradient boosting library, XGBoost, in which decision trees are generated sequentially. Credit card transactions are considered highly imbalanced in nature, making genuine cases very rare in comparison with fraudulent ones. The model consequently penalizes misclassifications of fraud cases with greater weightage. It accomplishes this through a cost-sensitive objective, which puts greater emphasis on false negatives (undetected frauds), which carry a heavier cost than false positives. All experiments were run on Python 3 with libraries such as pandas, numpy, scikit-learn, xgboost, imblearn, matplotlib, and seaborn.

4.1 Data preprocessing

European cardholders made credit card transactions during September 2013, which are recorded in this dataset. The dataset spans two days of transactions, which contain 492 fraudulent cases among 284,807 total transactions. The dataset shows a significant imbalance because the positive class representing fraud constitutes 0.172% of all transaction records. The study uses PCA-anonymized data to protect customer information during analysis. We cannot analyze the original features since they have been transformed into anonymous variables. The study keeps both "Amount" and "Time" features, which get standardized through the standard normalization method to match the scale of the other transformed features. The target variable has two classes since "0" stands for normal transactions and "1" represents fraudulent transactions.

Table 2. Statistical Summary of Amount and Time

	Time	Amount
Count	284807.000000	284807.000000
Mean	94813.859575	88.349619
Std	47488.145955	250.120109
Min	0.000000	0.000000
25%	54201.500000	5.600000
50%	84692.000000	22.000000
75%	139320.500000	77.165000
Max	172792.000000	25691.160000

The column chart displays the distribution of all dataset values. The two classes demonstrate a certain level of intersection in their data points. Figure 2 shows the column chart representation.

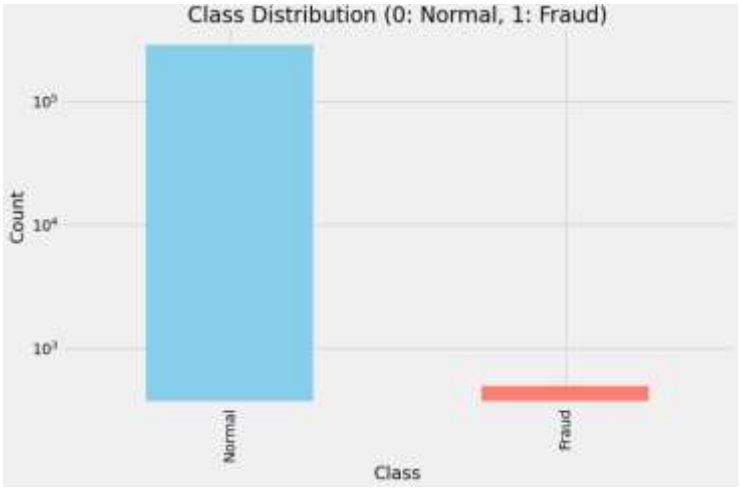

Figure 3. Class distribution

4.2 Train – Test Split

The training process achieves representativeness through an 80% training and 20% testing data split that uses stratified sampling. The stratification method maintains equal proportions of fraud and non-fraud cases between testing and training sets which becomes essential for imbalanced datasets.

4.3 Addressing Class Imbalance with SMOTE

The dataset contains an extreme class imbalance because fraudulent transactions make up only 0.2% of the entire dataset. SMOTE, which stands for Synthetic Minority Oversampling Technique, was applied only to the training set. SMOTE creates additional minority class samples by calculating new data points between original samples along with their closest neighbors. The method provides improved model capability to identify fraud boundaries through synthetic minority sample generation, which prevents model overfitting from repeated data duplication.

The resampling process created a training set with an equal number of legitimate and fraudulent transactions. The XGBoost classifier received scale_pos_weight = 1 since the classes were equally distributed.

4.4 Model Evaluation

The model evaluation utilized an untouched test set for performance measurement through standard classification metrics, including precision and recall, as well as F1-score,

ROC-AUC, and the confusion matrix. The selected metrics provide a complete assessment of the model's performance through its ability to find minority class instances (fraud) while minimizing false positives.

The evaluation process included visual representations such as ROC curves and confusion matrix heatmaps to enhance quantitative assessment and present clear classification patterns. The results received special attention for recall and false negatives because these metrics play a crucial role in detecting fraud.

4.5 Hyperparameter Tuning

Hyperparameter optimization is conducted using RandomizedSearchCV, a method from the scikit – learn library. Unlike GridSearchCV, which exhaustively searches over all parameter combinations, RandomizedSearchCV samples a given number of parameter settings from specified distributions. This method is more computationally efficient and often yields comparably optimal results, particularly when time and resources are limited.

In this study, the following hyperparameters were selected for tuning:

(1) Max_depth: Controls the maximum depth of the trees. Deeper trees may capture more complexity but are prone to overfitting.

(2) Learning_rate: Shrinks the contribution of each tree; smaller values make the model more robust but require more trees.

(3) N_estimators: The number of trees to fit

(4) Subsample: The fraction of training data used to build each tree helps prevent overfitting.

(5) Colsample_bytree: The fraction of features randomly sampled for each tree.

(6) Scale_pos_weight: Although SMOTE was used to balance the training data, this parameter was still included in some experiments to assess the effect of cost–sensitive adjustments.

A total of 50 parameter combinations were sampled using 3-fold cross-validation on the resampled training data. The optimization objective was to maximize the ROC – AUC score, as this metric captures the trade-off between the true positive rate and false positive rate, which is particularly relevant for imbalanced classification problems like fraud detection.

After tuning, the best set of hyperparameters was selected, and the model was retrained on the full resampled training set using these settings. The final model was then evaluated on the original test set using the same metrics as before.

This tuning process helped to refine the model's structure and learning behavior, resulting in improved detection capability and a better balance between recall and precision.

Evaluating model performance involves looking at five essential classification metrics to get a well-rounded view:

(1) Precision: This measures how many of the predicted fraud cases were actually correct. (2) Recall: This shows the percentage of actual fraud cases that the model successfully identified.

(3) F1-score: This is the harmonic mean of precision and recall, striking a balance between the two.

(4) ROC – AUC score: This metric illustrates the trade-off between the true positive rate and the false positive rate across various classification thresholds.

(5) Confusion matrix: This provides a summary of true positives, true negatives, false positives, and false negatives, giving a complete overview of classification results. These metrics offer both threshold-dependent and threshold-independent evaluations, allowing for a thorough assessment of model performance, especially in cases of imbalanced classification.

The study proposed a baseline model as shown in Figure 1.

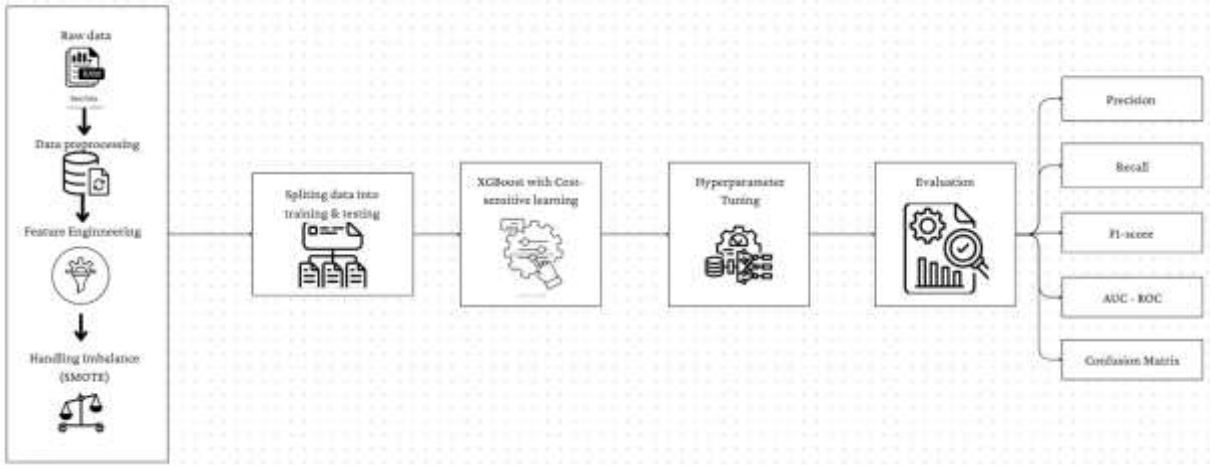

Figure 2. Proposed baseline model

5 Result

5.1 Experiment 1: Baseline model

To establish a point of comparison, the study began by training a baseline XGBoost model without implementing any class balancing techniques. The results that follow, which include the confusion matrix, classification report, and ROC curve, showcase how the model performed on the original imbalanced dataset. These findings act as a benchmark for assessing the enhancements brought about by the techniques that followed.

Figure 4. Confusion Matrix for XGBoost Classifier

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56864
1	0.87	0.80	0.83	98
accuracy			1.00	56962
macro avg	0.93	0.90	0.91	56962
weighted avg	1.00	1.00	1.00	56962

Figure 5. Performance of the XGBoost model

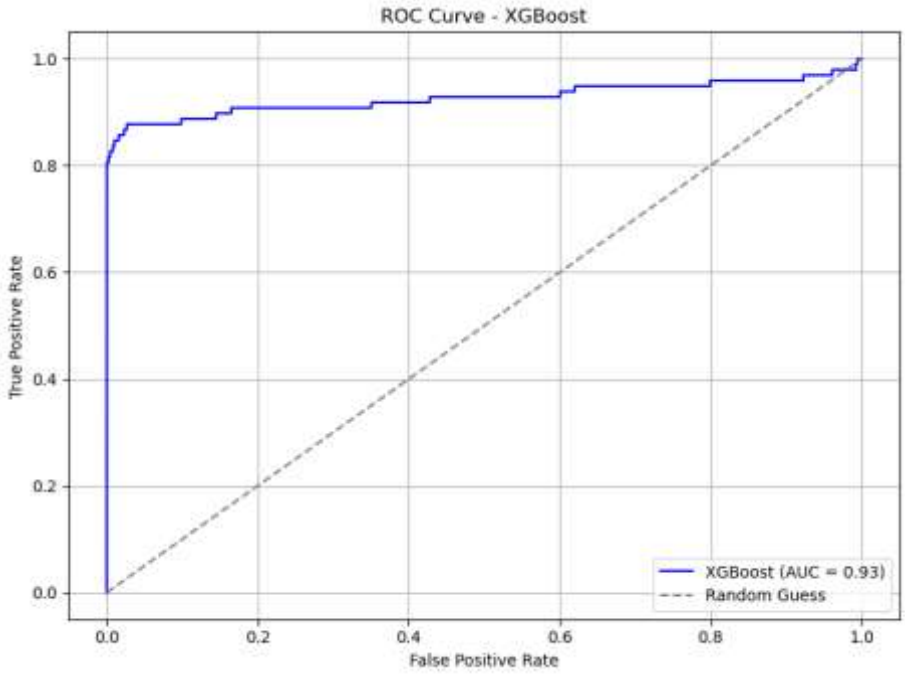

Figure 6. ROC Curve - XGBoost

The initial XGBoost model was set up without any cost-sensitive weighting mechanisms. Even with a significant class imbalance, the baseline model surprisingly performed quite well, achieving an impressive overall accuracy of nearly 100% and an AUC-ROC score of 0.93 (see Figure 4). The confusion matrix (Figure 2) reveals that 78 out of 98 fraudulent transactions were accurately identified, leaving just 20 false negatives. Additionally, the model showcased strong precision (0.87) and recall (0.8) for the minority class, leading to an F1 score of 0.83. These findings indicate that XGBoost's inherent boosting mechanism is already picking up on some characteristics of the rare class. However, those 20 missed fraud cases could still mean significant financial losses in real-world situations. Thus, there's a clear need for further enhancements to minimize false negatives. In the next section, we'll introduce a cost-sensitive version of the model aimed at improving the detection of the minority class by placing a heavier penalty on misclassifying fraud.

5.2 Experiment 2: XGBoost with cost sensitive and tuning hyperparameters

After using SMOTE to balance the training data and incorporating cost-sensitive learning, the research team retrained the XGBoost model and assessed its performance with the same metrics. The results presented below, which include the confusion matrix, classification report, and ROC curve, clearly show that the optimized model is much better at detecting fraudulent transactions while also reducing the number of false negatives.

Figure 7. Confusion matrix for Xboost – cost sensitive

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.69      0.87      0.77        98

    accuracy                           1.00     56962
   macro avg       0.84      0.93      0.88     56962
weighted avg       1.00      1.00      1.00     56962

ROC AUC Score: 0.9752791838457386
```

Figure 8. Performance of the Xgboost model with cost sensitive

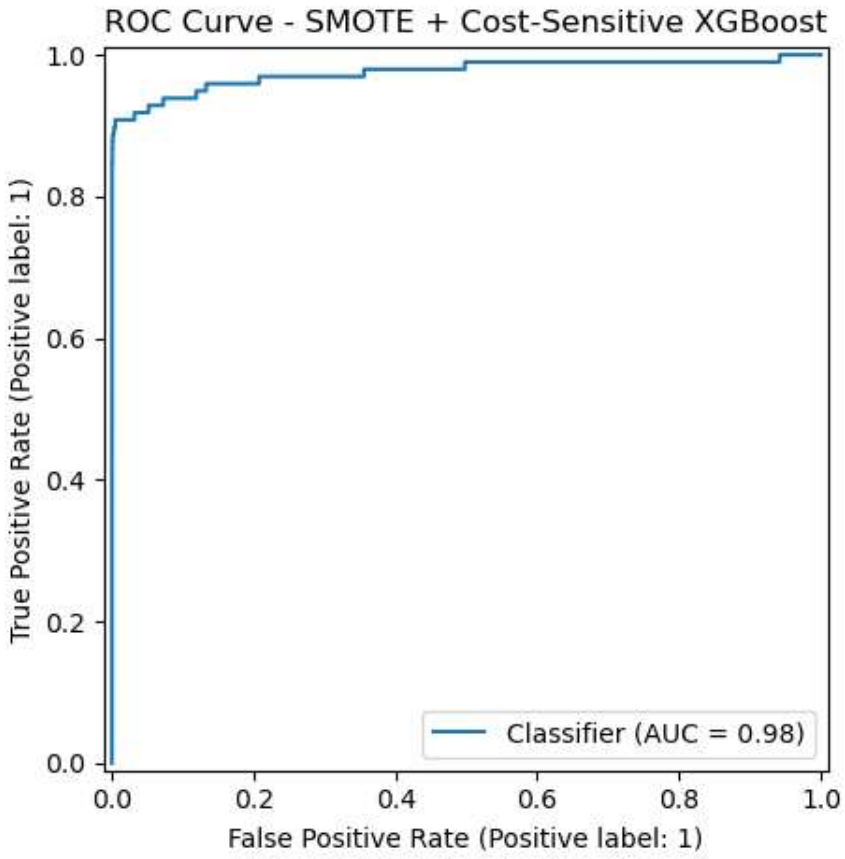

Figure 9. ROC Curve – Xgboost cost sensitive

The SMOTE and Cost-sensitive XGBoost model has some fantastic enhancements over the original XGBoost model, especially in identifying fraudulent transactions. For instance, the recall of the fraud category rose from 0.80 at the base to 0.87, indicating the number of fraud cases missing went from 20 to just 13. This improvement is very critical to fraud detection systems since missing fraud opportunities can make significant amounts of money. In addition, the ROC AUC value climbed from 0.95 to 0.98, showing an improved overall

capacity for distinguishing fraudulent and non-fraudulent transactions according to probability estimates. But with this improvement in recall was a sacrifice: precision dropped from 0.87 in the baseline to 0.69. As a result, the number of false positives rose from 12 to 39, which shows the enhanced sensitivity of the model to the minority class due to SMOTE. Despite this tradeoff, better recall combined with higher AUC indicates that the SMOTE-boosted model is better in scenarios where catching fraud is a more significant consideration than the periodic false positive. This makes it particularly well adapted to real-life financial applications, where the cost of losing fraud is far greater than the nuisance of struggling with a few false positives.

6 Evaluation & Comparison

To evaluate how well the proposed method works, we developed and compared two XGBoost models: one baseline model that was trained on the original imbalanced dataset, and another enhanced model that utilized a mix of SMOTE (Synthetic Minority Oversampling Technique) and cost-sensitive weighting. You can find a summary of the results in Table 2.

Table 2. Baseline vs Cost-Sensitive XGBoost: Performance Comparison

Metric	**Baseline XGBoost**	**Cost – sensitive XGBoost**
AUC – ROC	0.93	**0.98**
Fraud Precision	0.87	**0.69**
Fraud Recall	0.80	**0.87**
Fraud F1 – score	0.83	**0.77**
False Negatives	20	**13**
False Positives	12	**39**

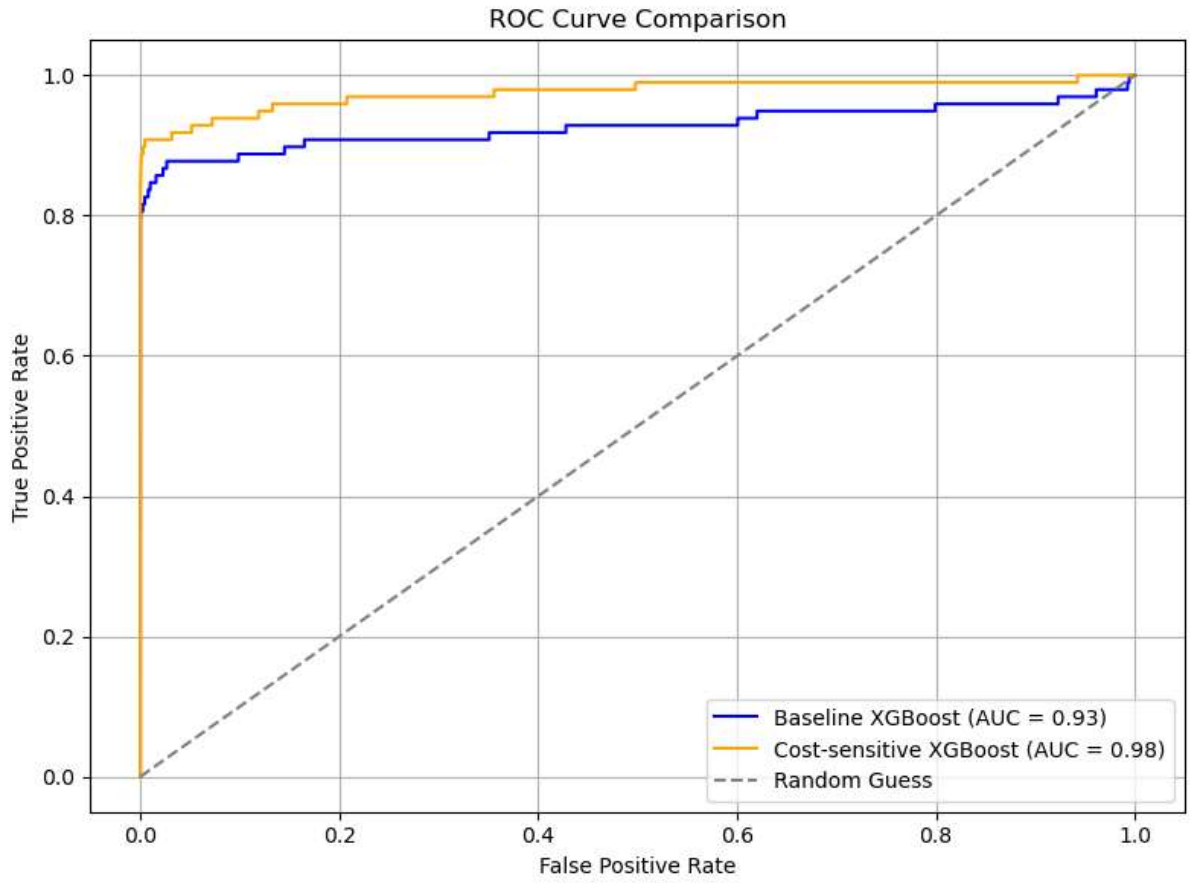

Figure 10. ROC Curve comparison

The underlying XGBoost model by itself performed very well overall with a good accuracy, having a precision of 0.87, a recall of 0.80, and an F1 score of 0.83 for detecting fraud. It was able to identify 78 out of 98 fraudulent transactions as fraudulent and had a ROC AUC of 0.93, which means that it performed well in discriminating between the classes. But it did miss 20 instances of fraud, and that's a problem in real-world applications because the cost of missed fraud is usually greater than the cost of false positives. On the other hand, the cost-sensitive XGBoost + SMOTE model improved recall to 0.87, detecting 85 instances of fraud out of 98 and minimizing false negatives to just 13. Such enhanced improvements are vital in fraud detection systems, where it is imperative to capture every instance of fraud. Also, the ROC AUC score increased to 0.98, indicating even higher class discrimination. All that said, the recall increase was nonetheless bought at the expense of a loss in accuracy, which decreased to 0.69 due to a rise in false positives from 12 to 39. The fraud class F1 score was 0.77, slightly worse than the baseline but reasonably so given the increased ability to detect. Briefly, the comparison indicates a common trade-off in class-imbalanced classification: enhanced recall generally occurs at the cost of precision. In the credit card fraud detection application, where false positive expenses can be high, this trade-off is

generally acceptable and even preferable. The enhanced model's higher recall and ROC AUC indicate its superior ability to detect fraud, making it an enhanced candidate for deployment in high-risk financial settings.

7 Conclusions and Future Work

This paper addressed the challenge of detecting fraudulent transactions from highly skewed credit card datasets using XGBoost, a strong gradient boosting algorithm. The base XGBoost model was highly effective in terms of overall performance, but was affected in detecting minority class instances (fraud) by data imbalance. To improve fraud detection, cost–sensitive learning and SMOTE (Synthetic Minority Oversampling Technique) were used together. This approach was intended to boost the sensitivity of the model towards the identification of rare fraud cases with reasonable accuracy. The SMOTE + cost–sensitive XGBoost model performed higher recall (0.87) and ROC AUC score (0.98) compared to the baseline model. This indicates that the enhanced model was better at identifying fraudulent transactions and avoiding classes. However, these enhancements came at the cost of accuracy, which declined to 0.69 as false positives rose. Nevertheless, the results show that assigning high recall is better suited to fraud detection systems, where a false alarm is less expensive than ignoring fraud. Overall, the findings verify the efficacy of employing a combination of data-level and algorithm–level approaches to improve fraud detection performance under imbalanced conditions.

While the current approach is promising, several avenues can be explored to further enhance model performance. First, sophisticated sampling techniques such as ADASYN (Adaptive Synthetic Sampling) or ensemble–based techniques such as BalancedBaggingClassifier can be investigated to increase minority class representation without substantially increasing false positives. Second, threshold tuning strategies may be applied post-training to find the best decision threshold balancing precision and recall based on business requirements. Apart from that, incorporating actual–world cost matrices with exact costs to false positives and false negatives may further render the model closer to financial risk. Finally, explainability is a significant factor; additional research should focus on incorporating model interpretation tools such as SHAP or LIME so that financial analysts can understand model predictions and gain confidence in the automated detection system.

References

[1] Y Bhasin, M. L. (2016). Combatting bank frauds by integration of technology: experience of a developing country. British Journal of Research, 3(3), 64-92.

[2] Olufemi, Bello & Bello, Oluwabusayo & Olufemi, Komolafe & Author, Corresponding. (2024). Artificial intelligence in fraud prevention: Exploring techniques and applications challenges and opportunities. 5. 1505 - 1520. 10.51594/csitrj.v5i6.1252.

[3] Świątkowska, J. (2020). Tackling cybercrime to unleash developing countries' digital potential. Pathways for Prosperity Commission Background Paper Series, 33, 2020-01.

[4] T. Bleach, "Juniper Research: Online Payment Fraud to Predicted to Exceed $362billion by 2028," The Fintech Times, Jun. 26, 2023. https://thefintechtimes.com/juniper-research-online-payment-fraud-to-predicted-to-exceed-362billion-by-2028/

[5] El Bouchti, A., Chakroun, A., Abbar, H., & Okar, C. (2017, August). Fraud detection in banking using deep reinforcement learning. In 2017 Seventh International Conference on Innovative Computing Technology (INTECH) (pp. 58-63). IEEE.

[6] Bello, O. A., & Olufemi, K. (2024). Artificial intelligence in fraud prevention: Exploring techniques and applications challenges and opportunities. Computer science & IT research journal, 5(6), 1505-1520.

[7] Abdallah, A., Maarof, M. A., & Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, *68*, 90-113.

[8] Dama, Krishna & Reddy, K Pavan Kalyan & Hrithik, K & Raheem, Dudekula & Vyshnavi,. (2024). Fraud Detection in Financial Transactions. 10.13140/RG.2.2.33977.99685.

[9] J. G. Lopez, "Fraud Detection in Financial Transactions: Types & How To Detect It," *Razorpay Learn*, Jan. 06, 2025. https://razorpay.com/learn/fraud-detection-in-financial-transactions/

[10] S. K. Hashemi, S. L. Mirtaheri and S. Greco, "Fraud Detection in Banking Data by Machine Learning Techniques," in IEEE Access, vol. 11, pp. 3034-3043, 2023, doi: 10.1109/ACCESS.2022.3232287.

[11] Noviandy, T. R., Idroes, G. M., Maulana, A., Hardi, I., Ringga, E. S., & Idroes, R. (2023). Credit card fraud detection for contemporary financial management using XGBoost-

driven machine learning and data augmentation techniques. *Indatu Journal of Management and Accounting*, *1*(1), 29-35.

[12] Zhang, R., Cheng, Y., Wang, L., Sang, N., & Xu, J. (2023). Efficient Bank Fraud Detection with Machine Learning. *Journal of Computational Methods in Engineering Applications*, 1-10.

[13] Minastireanu, E. A., & Mesnita, G. (2019). An Analysis of the Most Used Machine Learning Algorithms for Online Fraud Detection. *Informatica Economica*, *23*(1).

[14] Moreira, M. Â. L., Junior, C. D. S. R., de Lima Silva, D. F., de Castro Junior, M. A. P., de Araújo Costa, I. P., Gomes, C. F. S., & dos Santos, M. (2022). Exploratory analysis and implementation of machine learning techniques for predictive assessment of fraud in banking systems. *Procedia Computer Science*, *214*, 117-124.

[15] Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE.

[16] Gyamfi, N. K., & Abdulai, J. D. (2018, November). Bank fraud detection using support vector machine. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)* (pp. 37-41). IEEE.

[17] Chen, S., Goo, Y. J. J., & Shen, Z. D. (2014). A hybrid approach of stepwise regression, logistic regression, support vector machine, and decision tree for forecasting fraudulent financial statements. *The Scientific World Journal*, *2014*(1), 968712.

[18] M. Phankokkruad, "Cost-Sensitive Extreme Gradient Boosting for Imbalanced Classification of Breast Cancer Diagnosis," *2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Aug. 2020, doi: https://doi.org/10.1109/iccsce50387.2020.9204948.

[19] P. Gupta, S. Shukla, Vaishali Kikan, and A. Kumar, "Enhancing Fraud Detection in Credit Card Transactions using XGBoost and SMOTE: A Comparative Study," pp. 1–6, Jul. 2024, doi: https://doi.org/10.1109/icspcre62303.2024.10675080.

[20] Hajek, P., Abedin, M. Z., & Sivarajah, U. (2023). Fraud detection in mobile payment systems using an XGBoost-based framework. *Information Systems Frontiers*, *25*(5), 1985-2003.

[21] IBM, "XGBoost," Ibm.com, May 09, 2024. https://www.ibm.com/think/topics/xgboost

[22] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

[23] Zhang, Y., Trubey, P., & Hsieh, C. J. (2021). *XGBoost for fraud detection: Optimizing sparse transaction data processing*. arXiv preprint arXiv:2103.03863.

[24] Carcillo, F., Le Borgne, Y. A., Caelen, O., & Bontempi, G. (2018). Streaming active learning strategies for real-life credit card fraud detection. *Expert Systems with Applications, 91*, 235–245.

[25] M. Leo, S. Sharma, and K. Maddulety, "Machine Learning in Banking Risk Management: A Literature Review," *Risks*, vol. 7, no. 1, p. 29, Mar. 2019, doi: https://doi.org/10.3390/risks7010029.

[26] X. Hu *et al.*, "Cost-Sensitive GNN-Based Imbalanced Learning for Mobile Social Network Fraud Detection," *IEEE transactions on computational social systems*, pp. 1–16, Jan. 2024, doi: https://doi.org/10.1109/tcss.2023.3302651.