Fraud Detection in E-Commerce Card Transactions With Azure Machine Learning

Vineeth Ramakrishnan
Student ID: 23201606
Module:Cloud Machine Learning
M.Sc in Cloud Computing
National College of Ireland, Dublin, IRELAND
Email: x23201606@student.ncirl.ie

Sanjith Chokalingam Pillai
Student ID: 23194383
Module:Cloud Machine Learning
M.Sc in Cloud Computing
National College of Ireland, Dublin, IRELAND
Email: x23194383@student.ncirl.ie

Silver Stalan Inbaraj
Student ID: 22105441
Module:Cloud Machine Learning
M.Sc in Cloud Computing
National College of Ireland, Dublin, IRELAND
Email: x22105441@student.ncirl.ie

Arafah Lawal
Student ID: 20747315
Module:Cloud Machine Learning
M.Sc in Cloud Computing
National College of Ireland, Dublin, IRELAND
Email: x 20747315@student.ncirl.ie

***Abstract*—The global challenge of e-commerce fraud is so large that financial institutions across the world are losing $35 billion on an average every year due to such fraudulent activities. Credit card frauds result in annual financial losses that surpass billions of dollars for merchants and their consumers. This necessitates the early detection of fraud in financial institutions to decrease their monetary loss and stop derivative fraud patterns from emerging in the future. This research focuses on developing an Azure Cloud-based Machine Learning (AzureML) fraud detection system that is capable of analyzing the historical credit card transactions data and identify the fraudulent transactions from it. This is achieved using supervised ML algorithms such as logistic regression, Random Forest, XGBoost, LightGBM, CatBoost and AdaBoost for the classification of card transactions as fraud or legitimate. The methodology used covers data extraction and transformation, exploratory data analysis, model training and the assessment of the performance in terms of standard classification metrics, execution times and inference latency. We trained and evaluated our models on the IEEE-CIS Fraud Detection dataset from Kaggle. With an average inference time of 7ms per transaction, the XGBoost model provides ROC AUC score of 0.899, which also shows excellent balance of performance scalability. The findings of this work substantiate the dominance of the XGBoost and LightGBM models in performing e-commerce fraud detection with high F1 scores and acceptable inference times.**

Keywords— e-commerce, credit card frauds, machine learning, XGBoost, AzureML

I. Introduction

The frauds that constitute the field of e-commerce maintains its position as the primary international financial issue which leads to massive yearly expenses for both financial institutions and merchants. The surge in digital payment transactions caused by worldwide events also lead to an increase in the sophistication of fraudulent practices [1]. The rising transaction counts at e-commerce platforms make it progressively difficult to identify and prevent fraudulent activities without any well-designed fraudulent behavior detection infrastructure. Recent technological breakthroughs, especially in machine learning (ML) and cloud computing technologies have substantially accelerated fraud detection mechanisms [2-4].

This project aims at increasing the effectiveness of the conventional methods of fraud detection through cloud-based ML solutions [5]. This will be achieved by implementing a pipeline with steps to preprocess data, perform feature engineering, train and evaluate models, and deploying the models to a cloud environment for real-time model inference on transactional data and scale performance. This work is aimed at one primary task i.e., determining whether different algorithms can detect fraud in e-commerce transactions and predict activities which may potentially result in financial damage before it happens. In order to tackle this binary classification problem, we use various sophisticated supervised ML algorithms and compare their performance across various metrics such as accuracy, precision, recall, F1 score, ROC AUC, and training and inference times. The motivation for this study is to investigate fraudulent e-commerce transactions by the use of ML techniques in a cloud environment setting to evaluate the authenticity of the transaction. An appropriate e-commerce card transaction dataset that includes both transaction and identity information is used for running the experiments.

The main research objective is to improve the detection capacity of frauds with minimization of model capacity to shorter response times and low computation resources for real-time or near real-time detection in production environments. Thus, this research tries to answer the question: "Which proposed ML algorithms provide the optimal balance between detection accuracy and inference speed for e-commerce fraud detection?". For this research, publicly available e-commerce fraud transaction data is used as source of study. In this research, the proposed classification algorithms are developed and deployed on Microsoft Azure.

II. Related Work

Fraud detection in financial transactions, credit card usage and e-commerce platforms have been extensively researched in various domains. This challenge has been addressed by many studies using ML techniques where each of them has its own strengths and weaknesses. A most comprehensive study in this area was that of (Dal Pozzolo et al., 2017), where analysis was carried out for credit card fraud detection using ML. They noted the challenges of class imbalance and concept drift which accentuates the need for adaptive models. To handle the class imbalance issue, their research used undersampling techniques and attained decent performance using ensemble learning models. Some positive aspects of their work include solving actual fraud detection problems [6].

Nti and Somanathan, (2024) presented a scalable framework by combining Random Forest (RF) for feature selection and XGBoost for classification. The main advantage of their method was ensuring the balance of data through adaptive synthetic sampling (ADASYN) and engineered features to detect the transaction anomalies. Though their research was mainly driven by classification accuracy, it overlooked an important practical factor for real-time deployment scenarios [7]. (Han and Wai, 2024) performed a comparative analysis of boosting algorithms and discovered that XGBoost achieved best performance on card fraud detection compared to CatBoost, AdaBoost, Gradient Boosting, and LightGBM. This research provides invaluable insight into relative strengths of boosting algorithms on various metrics. However, this study did not consider model latency from the perspective of cloud deployments essential for e-commerce infrastructure [8].

Bagga et al., (2020) evaluated various ML approaches to detect credit card fraud with ensemble and pipelining methods that outperformed the traditional algorithms. They show how combining multiple models can achieve better detection performance on highly imbalanced datasets. However, they did not investigate the key factors like computational resources and inference frequencies in model deployments for e-commerce platforms [9]. (Nguyen et al., 2022) proposed a hybrid model by combining CatBoost and a Deep Neural Network (DNN) for card fraud detection in an innovative way to deal with the real-world problem in handling both loyal and first-landing customers where the old and new users are separated before different models are applied on each group. But the primary issue is that their deep learning (DL) model requires a significant number of resources and may not be feasible for real-time application within high volume transaction environments [10]. (Zeng et al., 2025) presented NNEnsLeG, a hybrid method that integrates ensemble learning and neural networks. They used transaction patterns, user behavior, and account correlations as a part of the feature engineering process to get a holistic view of the potential frauds. Although their approach is a significant advancement, it can be challenging for use on real-time inference models due to the complexity of their model architecture [11].

The learned aspects from reviewing these research works are a comparative study of ML models, the presence of real-world datasets with relevant data essential for transactional fraud analysis and improving prediction accuracies in a controlled environment. This review has also helped us identify some of the negative or unexplored aspects to the research domain like restricted model selection in some studies, lack of a comprehensive evaluation metric other than accuracy, and lack of focus on computational efficiency in a real-world scenario. Some studies also lacked statistical significance tests which question the validity of the observed differences in reported accuracy.

To overcome these research gaps, we compare several ML approaches to address those concerns to make them properly measurable and deployable. To deal with class imbalance, we implement techniques like under sampling to balance class distribution. Besides the accuracy, we present an evaluation standard with detailed metrics ranging from precision, recall, F1-score to AUC-ROC and PR curves. Our contribution also extends to developing a scalable, cloud-based solution using Azure ML services that strikes a balance between accuracy and computation efficiency, especially in measuring and optimizing inference times suitable for real-world deployments. This research, therefore, takes into account several factors and seeks to attain better results in fraud detection without neglecting the computational efficiency of the solutions for production environments.

III. Methodology

This research follows a systematic methodology of data acquisition, data preprocessing, exploratory data analysis (EDA), feature engineering, model training and evaluation in a cloud computing environment as shown in Fig.1. This section explains each of the selected approaches and the rationale behind each of the components that form the methodology.

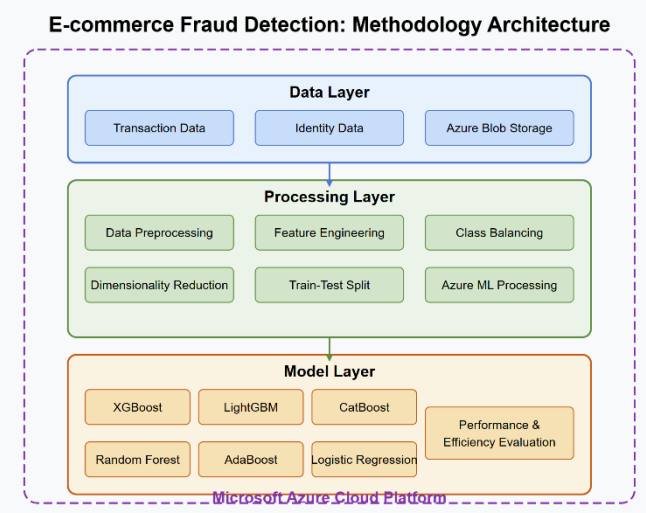

Fig. 1. Proposed E-Commerce Fraud Detection Methodology

A. Dataset Overview

This research uses a large dataset from Kaggle, namely the IEEE-CIS Fraud Detection challenge dataset[1]. This dataset consists of various features related to e-commerce activities such as transaction amount, device information, identity information and transactional behavior with the timestamp of each transaction. It contains both labeled training data and unlabeled test data as well as the binary target variable which indicates that a transaction is a fraud (1) or legitimate (0).

B. Data Acquisition and Preprocessing

The dataset used for the research comes from an IEEE-CIS Fraud Detection competition on Kaggle in which the research was conducted on the transaction data provided by Vesta Corporation, which is a leading e-commerce payment service provider. The one dataset which contains four base files: train_transaction.csv, train_identity.csv, test_transaction.csv, test_identity.csv. The dataset includes information of transaction, identity and device types. We stored it in the Azure Blob Storage and had a dedicated container named *frauddetection* for storage of the data. In

[1] https://www.kaggle.com/competitions/ieee-fraud-detection/data

fraud detection, data preprocessing is crucial, which includes handling missing values, encoding categorical variables, and normalizing numerical features to avoid losing information during iterations. Exploration of the data provided insights on features containing over 80% missing values and these large missing proportions led to the steps where such features were removed. Categorical variables were encoded using Label Encoding since one-hot encoding would create an inordinate number of dimensions.

C. *Exploratory Data Analysis*

EDA showed that a huge class imbalance existed in the dataset where fraudulent transactions accounted for a small percentage of all transactions. The challenge in training this model is in evading this imbalance that can possibly result in biased predictions. An analysis was also carried out as in Fig.2 to examine the transaction amount distribution and figure out its correlation with fraud status, where fraudulent transactions depict varied patterns different from legitimate ones. The analysis of the card types and product codes indicated differences in the fraud rate per category. A heatmap was created for the correlation matrix in order to identify highly correlated features for dimensionality reduction.

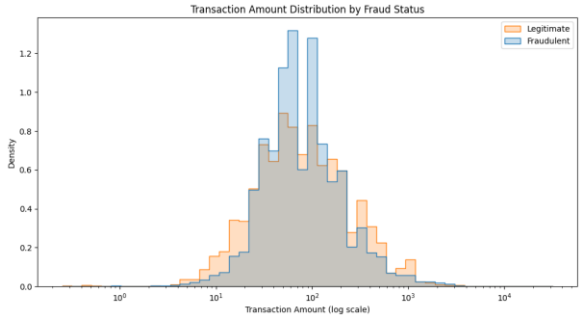

Fig. 2. Transaction Amount Distribution by Fraud Status

D. *Feature Engineering*

Feature engineering involves the creation of a new feature from an existing dataset that might help in detecting the fraudulent patterns. Also, dimensionality reduction was employed to reduce dimensionality without giving up the prediction capabilities of the data. A reduced feature space of 50 was obtained by using a method called Principal Component Analysis (PCA), where the reduced set still contains most of the information contained in the original data. This process relieves us from dimensionality curse, while also cutting down the computational cost in terms of model training and inference.

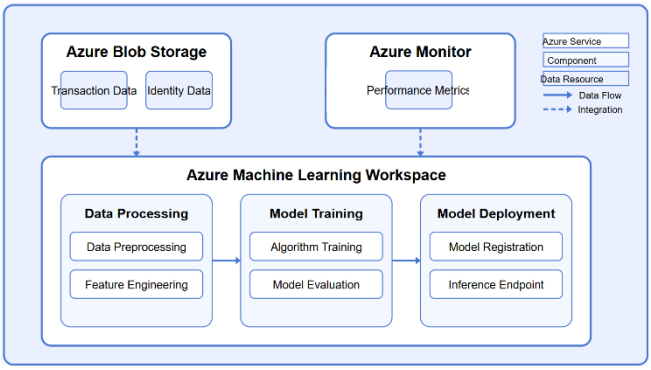

Fig. 3. Azure Cloud Implementation for Fraud Detection

E. *Class Imbalance Handling*

One of the main challenges in fraud detection is the class imbalance between fraudulent and legitimate transactions as most transactions are legitimate and just a small fraction of them turn out to be fraudulent. This problem was handled by Random Undersampling which simply removes samples from the majority class and balance the distribution. Using this technique, we reduce the training data size, make computation faster and bypass synthetic noise generated from oversampling method like SMOTE.

F. *Model Training*

Six machine learning algorithms, namely XGBoost, Random Forest, Logistic Regression, LightGBM, CatBoost, and AdaBoost were evaluated for fraud detection on the transactional dataset. These were chosen for their demonstrated efficacy in classification problems and different learning strategies. Logistic Regression is used as a statistical model that finds out the probabilities of class for a binary outcome. XGBoost was preferred for its speed and performance as well as its regularization capabilities. Random Forest is an ensemble method that uses the outputs from multiple decision trees trained, and the majority vote from those trees. LightGBM (Gradient Boosting Machines) is tree-based learning which facilitates distributed and efficient training. CatBoost automatically handles the categorical features, reducing the necessity for heavy data preprocessing. AdaBoost is an ensemble technique where multiple weak classifiers are combined together to provide a strong classifier and proves to be useful for previously misclassified instances.

These algorithms are compared and the most trustworthy method for e-commerce fraud detection can be determined. The preprocessing approaches with a balanced dataset was utilized for model training with all models trained on the same training data for a fair comparison. The commonly practiced approaches were used to configure the number of estimators, maximum depth, and learning rate. In order to compensate for any remaining class imbalance, logistic regression was set up with L2 regularization with balanced class weights. A fixed random seed was used across all the models to ensure reproducibility. To preserve the class distributions in the two sets, the dataset was split into stratified fashion into train (80%) and test (20%) sets.

G. *Azure Cloud Implementation*

The different services and components provided by Microsoft Azure were utilized to implement the fraud detection system on Azure are presented in Fig.3. The first step was to properly set up a cloud environment by creating an Azure subscription, and creating the appropriate resource groups, permissions etc. The project resources are managed through a resource group named "cmlprj" and Azure ML workspace 'fraud-detection' is provisioned with compute instances to centralize ML operations. For this project, an Azure storage account named 'cmldatasets' was set up with a container named 'frauddetection' to securely store dataset files in Azure Blob Storage. Then, the four main dataset files (train_transaction.csv, train_identity.csv, test_transaction.csv, and test_identity.csv) were uploaded to this container to act as a central data repository for the project.

IV. Results Evaluation

A. Model Performance Metrics

The model was evaluated in terms of fraud detection metrics such as accuracy, precision and recall. The performance of the model was evaluated using F1 Score, Area Under the Receiver Operating Characteristic Curve (ROC AUC), and Precision-Recall (PR) AUC. The F1-Score represents the harmonic mean of precision and recall, and ROC-AUC evaluates the model's capability of separating classes. PR-AUC can be useful with imbalanced datasets. The training time and inference time were the computational efficiency metrics critical to fraud detection systems that perform the task of decision making in real-time.

TABLE I. Model Performance Metrics Comparison

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC	PR AUC
XGBoost	0.818	0.836	0.791	0.813	0.899	0.914
LightGBM	0.809	0.833	0.775	0.803	0.887	0.902
CatBoost	0.788	0.815	0.747	0.779	0.862	0.882
Random Forest	0.787	0.814	0.746	0.778	0.865	0.887
AdaBoost	0.735	0.752	0.703	0.727	0.808	0.826
Logistic Regression	0.694	0.670	0.766	0.715	0.765	0.763

From Table-I, it can be inferred that XGBoost gives the best overall performance for the task of fraud detection as it scores highest in all the metrics on average. LightGBM had only slightly worse performance numbers in all categories but was close behind. The performance of CatBoost was comparable to Random Forest which had a slightly higher ROC-AUC, but lower precision and recall than CatBoost. However, both AdaBoost and Logistic Regression performed worse than other models. It was observed that Logistic Regression has the highest recall among all the models strengthening the claim that the model is effective in identifying fraudulent transactions, but at lower precision rate with more false positives. As seen in Fig. 4, XGBoost kept the highest ROC curve proving its strong discriminative capability under various thresholds. Precision-Recall curves additionally emphasized that XGBoost and LightGBM performed better than the rest, specifically by maintaining a high-precision for most recall levels which is important for fraud detection.

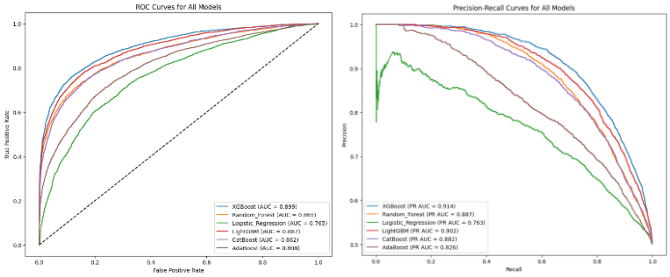

Fig. 4. ROC AUC and PR AUC Curves for All Models

B. Computational Efficiency Metrics

The training time study as presented in Table-II shows that LightGBM takes the least time (0.882s), followed by XGBoost with 1.633s. The training time of AdaBoost, Logistic Regression, and Random Forest are longer with compute duration ranging from 19.126s to 25.91s. Logistic Regression has the fastest inference time at 0.001 seconds per batch, followed by XGBoost at 0.008 seconds. CatBoost has the slowest inference time at 0.095 seconds. By analyzing the performance versus speed tradeoff, it is found that XGBoost has good performance and high speed which makes it suitable for the real-time fraud detection applications. CatBoost has a slower inference speed making it inapplicable to real-time applications, but its performance metrics are better than logistic regression whose inference speed is the best, although at the cost of performance metrics.

TABLE II. Model Computational Efficiency Metrics

Model	Training Time (s)	InferenceTime (s)
XGBoost	1.633	0.008
LightGBM	0.882	0.017
CatBoost	1.934	0.095
Random Forest	19.126	0.070
AdaBoost	25.910	0.084
Logistic Regression	23.145	0.001

A ranking analysis was conducted to establish the relation and importance in balancing between performance and computational efficiency evaluation by ranking across four key dimensions of training speed, inference speed, F1 score and ROC AUC for each model. As seen in Fig.5, XGBoost turned out to have the best average ranking (1.5) and good performance metrics together with an acceptable computing speed. With an average ranking of 2.0, LightGBM was another great performer with excellent training speed, and also strong performance metrics. Logistic Regression (4.25) has mixed profile, with the best possible inference speed, but relatively worse performance metrics, whereas Random Forest (3.75) and CatBoost (4.0) were found to be mediocre. AdaBoost turned out to have the lowest overall score (5.5) with worser performance and poor computational efficiency.

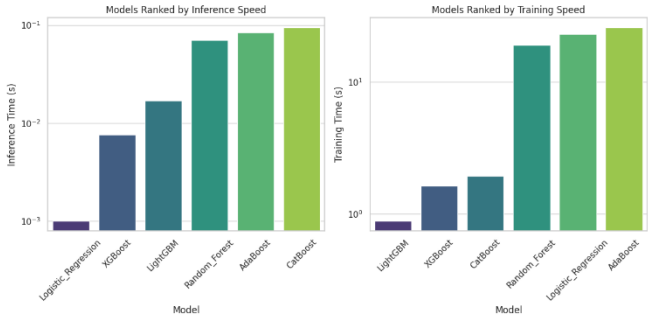

Fig. 5. Model Ranking Summary – Inference and Training Speeds

C. Model Inference on Test Dataset

An independent test dataset was used to examine the practical deployment capabilities of the trained models by performing inference. The test transactions were fed through each model, which was loaded from its serialized state from AzureML registry to predict fraud probabilities. The inference results for all models are presented in Table III.

TABLE III. Model Inference Results on Test Dataset

Model	Average Inference Time (s)	Sample Predictions (Fraud Probability)
XGBoost	0.007466	[0.402, 0.622, 0.256, 0.779, 0.211]
Random Forest	0.070771	[0.349, 0.546, 0.376, 0.566, 0.251]
Logistic Regression	0.000964	[0.485, 0.794, 0.413, 0.456, 0.489]

LightGBM	0.015664	[0.357, 0.588, 0.329, 0.593, 0.183]
CatBoost	0.095969	[0.417, 0.678, 0.369, 0.709, 0.291]
AdaBoost	0.088079	[0.483, 0.509, 0.484, 0.512, 0.483]

Finally, the computational efficiency observation in the model evaluation phase is verified by the inference results. However, in terms of inference time, Logistic Regression surpassed XGBoost with the inference time of 0.000964s while XGBoost took 0.007466s. The slowest inference was again reported by CatBoost (0.095969). At this stage, we can see how each model in assigning fraud probabilities. AdaBoost had lower discrimination in the probabilities in terms of 0.5 threshold (i.e. more probabilities were clustered near the interval (0.4, 0.6)) making it unprecise, while XGBoost and CatBoost had more pronounced differentiation between fraudulent and legitimate transactions.

V. Conclusions and Future Work

In this research, six different supervised ML algorithms were selected and implemented for e-commerce fraud detection and evaluated based on the performance metrics and computation efficiency. The most promising approaches are XGBoost and LightGBM where they are both quite accurate, but with data characteristics more fitting for near real-time inference speed suitable for real-world fraud detection systems. Based on this experience, the cloud-based implementation of Microsoft Azure helped to manage these tradeoffs by offering the requisite infrastructure to train complex models and to deploy them in an efficient manner.

This research also highlights the precision-recall tradeoff as a dimension for evaluating fraud detection models. The cloud-based architecture demonstrates its scalability and flexibility to detect the fraud effectively as expected from a modern fraud detection system. There are many approaches for future work including implementing those approaches using deep learning models, developing automated feature engineering, building ensemble models, and developing real-time model monitoring and updating models to address concept drift as new fraud patterns emerge.

In conclusion, a detailed comparison of tradeoffs between performance and efficiency for various ML algorithms for e-commerce fraud detection has been implemented in Azure ML cloud environment successfully. XGBoost and LightGBM are the promising ones with respect to both their high detection performance and reasonable computational efficiency for real-time model inference.

VI. Contributions

This research project was meant to be collaborative effort across various stages of the research with specific contributions involving each team member as detailed in Table-IV.

TABLE IV. Individual Project Contributions

Project Modules	**Vineeth**	**Sanjith**	**Silver**	**Arafah**
Research Background and Objectives	✓	✓		
Literature review	✓			✓
Azure Infra setup	✓		✓	
EDA		✓	✓	
Data preprocessing		✓	✓	
Feature engineering		✓	✓	
Model (XGBoost, LightGBM)	✓	✓		
Model (CatBoost, Random Forest)	✓			✓
Model (LR, AdaBoost)			✓	✓
Performance evaluation	✓	✓	✓	✓
Visualization development		✓	✓	
Report preparation	✓	✓	✓	✓
Project coordination	✓			

References

[1] B. Sturc, T. Gurova, and S. Chernov, "The specifics and patterns of cybercrime in the field of payment processing," *International Journal of Criminology and Sociology*, vol. 9, pp. 2021–2030, Apr. 2022, doi: 10.6000/1929-4409.2020.09.237.

[2] A. Ali *et al.*, "Financial Fraud Detection Based on Machine Learning: A Systematic Literature review," *Applied Sciences*, vol. 12, no. 19, p. 9637, Sep. 2022, doi: 10.3390/app12199637.

[3] A. Mutemi and F. Bacao, "E-Commerce fraud detection Based on Machine Learning Techniques: Systematic Literature review," *Big Data Mining and Analytics*, vol. 7, no. 2, pp. 419–444, Apr. 2024, doi: 10.26599/bdma.2023.9020023.

[4] G. Manoharan, S. D. N. H. Ali, M. Sathe, A. Karthik, A. Nagpal, and A. Sidana, "Fraud Detection in E-commerce Transactions: A Machine Learning perspective," *2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 1–5, May 2024, doi: 10.1109/accai61061.2024.10601813.

[5] B. Stojanović and J. Božić, "Robust financial fraud alerting system based in the cloud environment," Sensors, vol. 22, no. 23, p. 9461, Dec. 2022, doi: 10.3390/s22239461.

[6] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit Card Fraud Detection: a realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3784–3797, Sep. 2017, doi: 10.1109/tnnls.2017.2736643.

[7] I. K. Nti and A. R. Somanathan, "A scalable RF-XGBOOST framework for financial fraud mitigation," *IEEE Transactions on Computational Social Systems*, vol. 11, no. 2, pp. 1556–1563, Oct. 2022, doi: 10.1109/tcss.2022.3209827.

[8] S. S. Han and K. K. Wai, "A Performance Analysis of Boosting Algorithms for the Identification of Card Fraud," *IEEE Conference on Computer Applications (ICCA)*, pp. 1–6, Mar. 2024, doi: 10.1109/icca62361.2024.10532990.

[9] S. Bagga, A. Goyal, N. Gupta, and A. Goyal, "Credit Card Fraud Detection using Pipeling and Ensemble Learning," *Procedia Computer Science*, vol. 173, pp. 104–112, Jan. 2020, doi: 10.1016/j.procs.2020.06.014.

[10] N. Nguyen et al., "A proposed model for card fraud detection based on CatBoost and deep neural network," IEEE Access, vol. 10, pp. 96852–96861, Jan. 2022, doi: 10.1109/access.2022.3205416.

[11] Q. Zeng, L. Lin, R. Jiang, W. Huang, and D. Lin, "NNEnsLeG: A novel approach for e-commerce payment fraud detection using ensemble learning and neural networks," *Information Processing & Management*, vol. 62, no. 1, p. 103916, Oct. 2024, doi: 10.1016/j.ipm.2024.103916.