Application of machine learning for predicting fraudulent anomalies in financial transactions

Victoria Vysotska[1,2,†], Dmytro Uhryn[3,†], Oleksii Iliuk[4,†], Yuriy Ushenko[3,*,†] and Vasyl Yatsyshyn[5,†]

[1] *Information Systems and Networks Department, Lviv Polytechnic National University, Stepan Bandera Street 12 79013 Lviv, Ukraine*

[2] *University of Twente, Drienerlolaan 5, 7522 NB, Enschede, The Netherlands*

[3] *Yuriy Fedlovyvh Chernivtsi National University, Kotsiubynskoho Street 2 58012, Chernivtsi, Ukraine*

[4] *Senior IT system&business analyst, Temabit, 02000 Kyiv, Ukraine*

[5] *Ternopil Ivan Puluj National Technical University, Ruska Street 56, 46025, Ternopil, Ukraine*

Abstract

The article is devoted to the topical problem of detecting fraudulent anomalies in financial transactions using machine learning methods. In the context of rapid digital transformation of financial systems and growth in transaction volumes, traditional methods of fraud detection are becoming ineffective, which highlights the urgent need to implement automated and adaptive solutions. The research is based on a step-by-step approach that includes data preparation and processing, building and training classification models, and evaluating their effectiveness. A comparative analysis of seven popular machine learning algorithms was conducted: linear regression, decision trees, random forest, neural networks, gradient boosting, XGBoost, and SVC. The key findings of the study showed that ensemble methods demonstrate the highest effectiveness in detecting fraud: Random Forest, Gradient Boosting, and XGBoost proved to be the most suitable for fraud detection tasks, demonstrating consistently high results. This is especially important given the typical class imbalance (a small number of fraudulent transactions compared to legitimate ones) in real financial data. The effectiveness of the models significantly outperforms the other algorithms considered, indicating their ability to detect complex, non-obvious patterns in the data. The critical importance of correctly configuring model hyperparameters and accounting for class imbalance to achieve maximum accuracy and completeness in detecting fraudulent transactions has been confirmed. This avoids overfitting on the dominant class and increases the system's sensitivity to rare but important fraudulent cases. The practical significance of the study lies in the fact that the proposed approach allows financial institutions to significantly improve operational efficiency, minimize financial losses, and strengthen customer trust. The implementation of such systems provides comprehensive and adaptive protection of the financial system in today's dynamic digital environment. The results of the study confirm the effectiveness of machine learning as a powerful tool for combating financial fraud.

Keywords

Machine learning, financial fraud, anomaly prediction, ensemble methods, Random Forest, Gradient Boosting, XGBoost, class imbalance, financial transactions

1. Introduction

In today's world, financial systems are undergoing rapid digital transformation, marked by growing transaction volumes and increasingly complex financial operations. These changes, however, bring heightened risks of fraudulent activity, posing a serious threat to both financial institutions and their clients. Fraud in the financial sector is becoming more sophisticated, leveraging modern technologies to bypass traditional security measures.

[*] *AISSLE-2025: International Workshop on Applied Intelligent Security Systems in Law Enforcement, October, 30–31, 2025, Vinnytsia, Ukraine*

[1*] Corresponding author.

[†] These authors contributed equally.

victoria.a.vysotska@lpnu.ua (V. Vysotska); d.ugryn@chnu.edu.ua (D. Uhryn); olexiyilyukm@gmail.com (O. Iliuk); y.ushenko@chnu.edu.ua (Y. Ushenko); yacyshyn@tntu.edu.ua (V. Yatsyshyn)

0000-0001-6417-3689 (V. Vysotska); 0000-0003-4858-4511 (D. Uhryn); 0000-0002-0904-3045 (O. Iliuk); 0000-0003-1767-1882 (Y. Ushenko); 0000-0002-5517-6359 (V. Yatsyshyn)

According to the Association of Certified Fraud Examiners (ACFE), global organizations lose over 5% of their annual revenue to financial fraud. The 2024 report highlights that 53% of fraud cases were linked to factors stemming from the COVID-19 pandemic, and for the first time since 2016, the average loss per case increased. Criminals are increasingly using cryptocurrencies to cover their tracks and often operate in regions with weaker financial oversight. On average, a typical fraud scheme lasts around 12 months before detection, underscoring the urgent need for more effective monitoring tools.

ITraditional fraud detection methods—relying on static rules and manual analysis—are insufficient for today's challenges, driving interest in machine learning and AI for automated, adaptive real-time fraud detection. This paper develops a machine learning approach for identifying financial anomalies, aiming to create an effective system that detects suspicious transactions early to minimize losses, reputational harm, and legal costs. Automation boosts efficiency, ensures regulatory compliance, and builds client trust.Ultimately, advanced analytics provide comprehensive financial system protection, bolstering resilience in a dynamic digital landscape.

2. Related works

Digital transformation has reshaped financial transactions, making them faster, more convenient, and accessible via innovations like digital banking, mobile payments, cryptocurrencies, and fintech, which boost interconnectivity. Yet, it amplifies risks, especially financial fraud, threatening institutions' stability and customer trust.

Fraud includes identity theft, phishing, card fraud, money laundering, etc. Research [1-3] highlights growing challenges in countering it amid surging transaction volumes and evolving tactics.

Traditional methods—rule-based systems, risk filters, manual reviews—rely on fixed criteria (e.g., amount, location) but show limited effectiveness [3-7] in dynamic settings, often detecting fraud post-fact and causing losses. Figure 1 illustrates the rising payment fraud attempts over time.

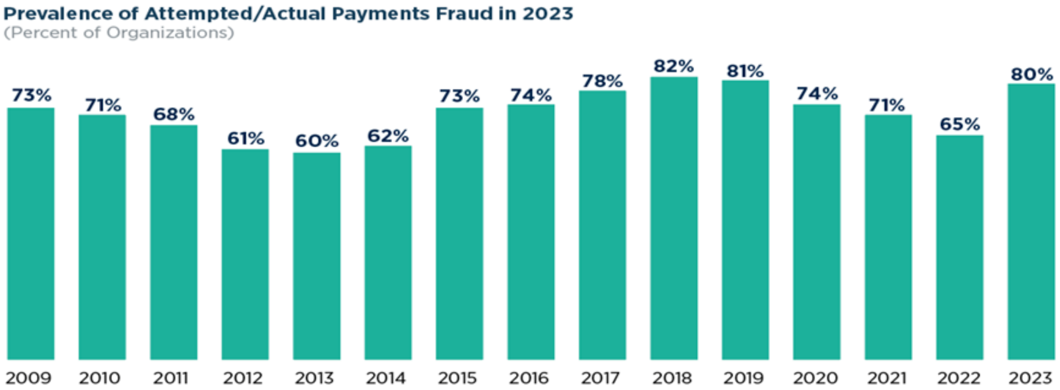


Figure 1: The spread of payment fraud attempts over the years

The emergence of modern technologies—particularly machine learning (ML), artificial intelligence (AI), and big data—has opened new opportunities for detecting and preventing complex fraud schemes. ML algorithms enable real-time transaction analysis, anomaly detection, and adaptive learning based on changing patterns of malicious behavior [8–10]. Behavioral analytics plays a key role as well, allowing institutions to build customer profiles and identify deviations from typical interaction patterns [11–13].

In addition, technologies such as blockchain and distributed ledger technology (DLT) enhance the transparency of financial processes and make fraudulent activities more difficult to execute by ensuring that data entries cannot be altered or forged without leaving a trace [14, 15].

Comprehensive, ready-to-deploy fraud detection systems are rarely available in the public domain. Most fraud detection models are developed under commercial contracts tailored to specific financial institutions or enterprises, using confidential, business-specific training data that restricts applicability in academic or open-source projects.Public alternatives are research models on platforms like Kaggle, using open datasets (e.g., Credit Card Fraud Detection) for full development —from data cleaning/normalization to model building and visualization. Valuable for evaluating algorithms in controlled environments, but challenging to integrate into real-world processes due to variances in data scale, structure, and dynamics.Academic literature reflects a growing interest in applying AI methods to combat financial fraud. Studies [16] explore the use of neural networks, decision trees, naïve Bayes classifiers, and ensemble models. Some researchers [17,18] have proposed using recurrent neural networks (RNNs) to process transaction sequences, achieving significantly improved results compared to traditional algorithms [19]. There is also an increasing emphasis on the need for continuous model adaptation to evolving fraud patterns [20,21]. To address this, hybrid systems combining supervised and unsupervised learning are being proposed, enabling the detection of new, previously unclassified types of fraud.According to analytical reports (ACFE, 2022; PwC, 2023), fraud detection is becoming a multidisciplinary challenge that spans not only risk management and IT but also marketing, customer service, and strategic management. Modern organizations must integrate risk analytics into all business processes, building collaborative teams that bring together experts from various fields.The cost of financial fraud is typically assessed by calculating both direct and indirect losses: fraudulent transaction losses, software and tool expenses, analyst salaries, legal fees, and opportunity costs resulting from diminished customer trust. A visualization of this approach is presented in the diagram below (Figure 2).

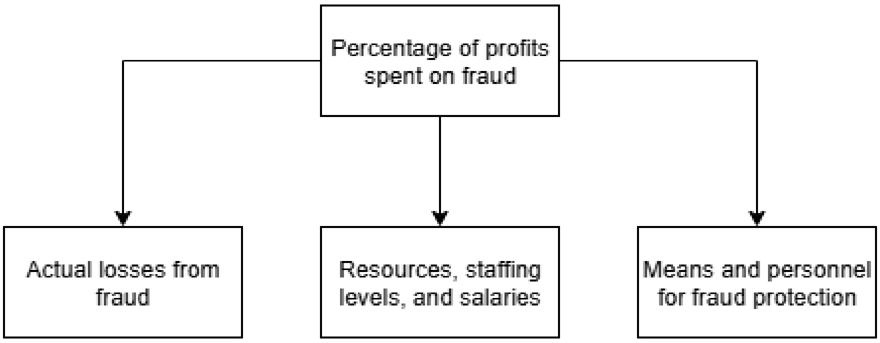

Percentage of profits spent on fraud
Actual losses from fraud
Resources, staffing levels, and salaries
Means and personnel for fraud protection

Figure 2: Scheme for calculating losses from fraud

This diagram illustrates a comprehensive assessment of the financial impact of fraud on an organization, capturing the various components that contribute to the overall cost:

1. Direct fraud losses. These are the actual financial damages incurred due to fraudulent activity, such as unauthorized withdrawals, transaction manipulation, or data theft.
2. Human resources and salary costs. Expenses related to the personnel involved in fraud detection, investigation, and prevention, including their salaries, working hours, and other risk management-related costs.
3. Fraud protection tools and staffing. Investments in software, technical security systems, analytical platforms, and Cybersecurity and compliance specialists.
4. Profit margin allocated to fraud-related costs. A portion of the company's overall profit that must be redirected toward covering fraud-related expenses—both direct and indirect.

The diagram highlights that the cost of fraud extends far beyond direct financial losses—it also includes expenditures on staff, technology, prevention tools, and lost profits. For this reason, organizations are increasingly motivated to implement efficient fraud detection and prevention systems to minimize the total financial burden.

3. Problem formulation

The aim of this study is to develop an effective system for detecting fraudulent transactions using machine learning methods. To achieve this, it is necessary to perform a set of tasks that cover the stages of data preparation, model development and tuning, as well as evaluation of their performance.

1. Data preparation and preprocessing. The first step is to create a high-quality input dataset, which includes several stages:

- Data cleaning. This involves removing duplicates, eliminating incorrect entries, and handling missing values.
- Normalization of numerical features. For example, scaling transaction amounts to ensure stable performance of the algorithms.
- Encoding categorical variables. This is achieved through techniques such as one-hot encoding or label encoding.
- Feature engineering. At this stage, new features are generated, including temporal, geographical, or behavioral patterns that may be informative for classification.

2. Building machine learning models. The next stage is the selection and configuration of models capable of recognizing anomalous transactions. Since the problem is a classification task, several approaches will be tested:

- Logistic regression.
- Decision trees.
- Ensemble methods. This includes Random Forest, Gradient Boosting, and XGBoost.
- Artificial neural networks.

Because the dataset is often highly imbalanced, with fraudulent transactions representing only about 1% of the total, special training strategies are required. These include adjusting class weights, oversampling, or undersampling. In addition, hyperparameter optimization will be performed to improve the accuracy and stability of the models.

3. Evaluation of model performance. The effectiveness of the models will be measured using several metrics: Accuracy, Recall, Precision, F1-score, ROC-AUC. To provide deeper insights, the models will be compared on datasets with different class distributions, including 50/50, 99/1, and 83/17. This will allow for assessment of system robustness under varying levels of imbalance and adaptability to real-world scenarios.
4. Data selection and preparation for experimentation. Since access to real financial transaction data is limited, publicly available datasets from Kaggle will be used. The choice of datasets will be guided by the availability of fraud labels, the structure of the data fields (categorical and numerical features), and sample size. After being downloaded, the data will undergo preprocessing and only then will be used for training the models.

4. Methods and materials

The table below summarizes key traits of the five algorithms, including convergence speed (iterations to 90% optimal on CEC 2017 functions) and computational complexity (operations per iteration for 100 agents). These aid in selecting algorithms for specific tasks.

In this study, a machine learning method detects fraudulent anomalies in financial transactions to minimize losses for institutions, businesses, and individuals. It is implemented stepwise using modern Python libraries.

Stage 1: Data preparation. Load dataset with pandas; clean by removing missing values, duplicates, and irrelevant attributes. Result: structured DataFrame.

Stage 2: Data scaling and normalization. Normalize numerical features (e.g., Amount) via RobustScaler from sklearn.preprocessing (median-based, outlier-resistant). Essential for scale-sensitive models like logistic regression, SVMs, neural networks, and gradient descent. Remove unsuitable features (e.g., ID, geography, non-informative fields).

Stage 3: Training/test set formation. Split data 80/20 using train_test_split (test_size=0.2, random_state=42, stratify=y) to prevent overfitting, tune hyperparameters, evaluate on unseen data, and preserve class balance.

Stage 4: Model construction and training. Binary classification (fraud/non-fraud) is applied. Models (e.g., logistic regression, decision trees, random forest, gradient boosting) are tuned individually with parameters like tree count, depth, or learning rate, balancing quality and resources.

Stage 5: Performance evaluation. Assess models using confusion matrix (TP, FP, FN, TN), classification report (precision, recall, F1-score), AUC-ROC (class separation at varying thresholds), and accuracy (correct predictions share; supplementary due to imbalance).

The methodology is adaptable to other fraud detection datasets.

The study evaluated seven ML algorithms for financial anomaly detection; each has strengths and limitations suiting specific data types.

1. Linear Regression. Despite its simplicity, this model can be applied to binary classification. It allows for class weighting, which is important for imbalanced datasets. The best results were shown by saga, while liblinear was more effective for smaller samples. The model is limited in its ability to capture complex nonlinear relationships and requires prior feature scaling.
2. Decision Tree. An interpretable and fast model that does not require feature scaling. In the study, it demonstrated a tendency toward overfitting, which is why depth was restricted to five levels, and thresholds for minimum node splits and samples per leaf were introduced. The model is sensitive to changes in data and performs worse than ensemble methods in terms of accuracy.
3. Random Forest. An ensemble approach based on decision trees that significantly reduces the risk of overfitting. It showed consistently high results across all datasets. The model used 50 trees with a maximum depth of 10 levels and a minimum of 10 samples per leaf. Another advantage is the ability to determine feature importance.
4. Neural Network (MLP Classifier). A multilayer model that performs well with complex and high-dimensional data. Its main advantages are flexibility and the ability to model complex dependencies. However, the model requires careful tuning of its architecture, considerable computational resources, and a large amount of training data. It is also difficult to interpret.
5. Gradient Boosting. An ensemble method with sequential tree training that corrects the errors of previous models. It demonstrated high accuracy with moderate depth (up to 3) and number of trees (100). The learning rate was set at 0.1 to balance speed and performance. A key advantage is robustness to imbalanced data due to class weight support.
6. XGBoost. An optimized version of gradient boosting with high performance. It supports handling missing values and includes built-in regularization that reduces the risk of overfitting. It was applied with the same parameters as Gradient Boosting and confirmed its effectiveness across different datasets.
7. Support Vector Classifier (SVC). One of the most accurate but also the most resource-intensive algorithms. It performs best on imbalanced datasets, where class inequality can be

compensated by class weights. Training took a significant amount of time (up to an hour), but the model demonstrated strong capability in separating complex classes.

These results confirm the suitability of ensemble methods (Random Forest, Gradient Boosting, XGBoost) for fraud detection tasks and highlight the importance of parameter tuning and class imbalance handling to improve model accuracy.

Data splitting into training and test sets is a fundamental stage in the machine learning process, as it allows for an objective evaluation of the model's ability to generalize to new, unseen data. This approach divides the dataset into two parts.

1. The training set is used to build the model and usually represents 70–80% of the data, providing the model with a wide range of examples to learn transaction patterns of different types. The larger the training set, the more patterns, both legitimate and fraudulent, the model can capture, which increases its predictive effectiveness.
2. The test set consists of the remaining 20–30% of the data, which is not used during training. It serves to evaluate the performance of the model on new examples, allowing for an assessment of its generalization ability. This corresponds to the supervised learning concept, where the model is first "trained" and then "tested" on an independent set.

When splitting data, parameters include: test_size=0.2 (20% for testing), stratify=y (ensures proportional class representation in train/test sets, crucial for imbalances), random_state=42 (fixes randomness for reproducibility and bias prevention). These minimize overfitting, where models perform well on training data but poorly on new data.

Evaluation uses the classification report to summarize key metrics; Precision is the proportion of correctly predicted positives among all predicted positives.

A high precision value indicates a small number of false positives. It is calculated using the formula:

$$Pre\,cision = \frac{TP}{TP + TF} \tag{1}$$

where *TP* represents true positives and *FP* represents false positive predictions. Recall (sensitivity) is a metric that reflects the model's ability to detect all actual positive cases. It is calculated as the proportion of correctly classified positive results among all real positive examples. A high recall value indicates a small number of false negative predictions. Formally, it is computed using the formula:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where *TP* represents true positives and *FN* represents false negatives. The F1-score is the harmonic mean between precision and recall, allowing a balance between these two metrics. It is particularly useful in cases of imbalanced data, where it is important to account for both false positive and false negative predictions. It is calculated using the following formula:

$$F1 = \left(\frac{2}{recall^{-1} + precision^{-1}} \right) = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3}$$

Support: Number of actual instances per class in the test set; aids in interpreting precision and recall relative to class sizes.

ROC-AUC: Measures model's class distinction at varying thresholds. ROC curve plots TPR vs. FPR as threshold changes. AUC: 1.0 = perfect, 0.5 = random guessing, <0.5 = worse than random.

Accuracy: Proportion of correct classifications overall. Simple but misleading in imbalanced data (e.g., predicting 95% majority class yields 95% accuracy but fails to detect minority):

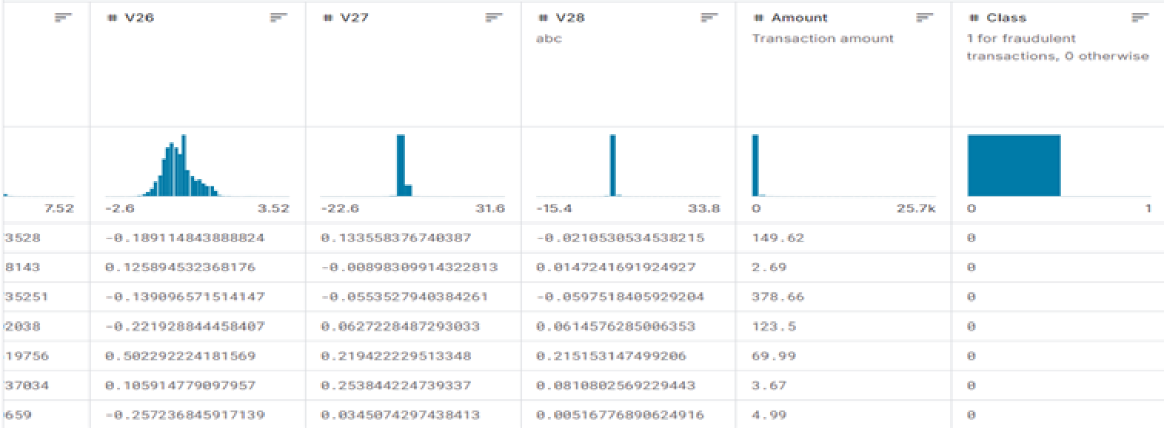
$$Accuracy = \frac{TP + TN}{TP + TN + TP + FN} \tag{4}$$

where TP = true positives, TN = true negatives, FN = false negatives. Metrics like confusion matrix, AUC-ROC, and accuracy enable comprehensive model evaluation, highlighting strengths and limitations. This supports informed selection of optimal methods, forming the basis for effective fraud detection systems that minimize risks for businesses and users.

5. Analysis of the database

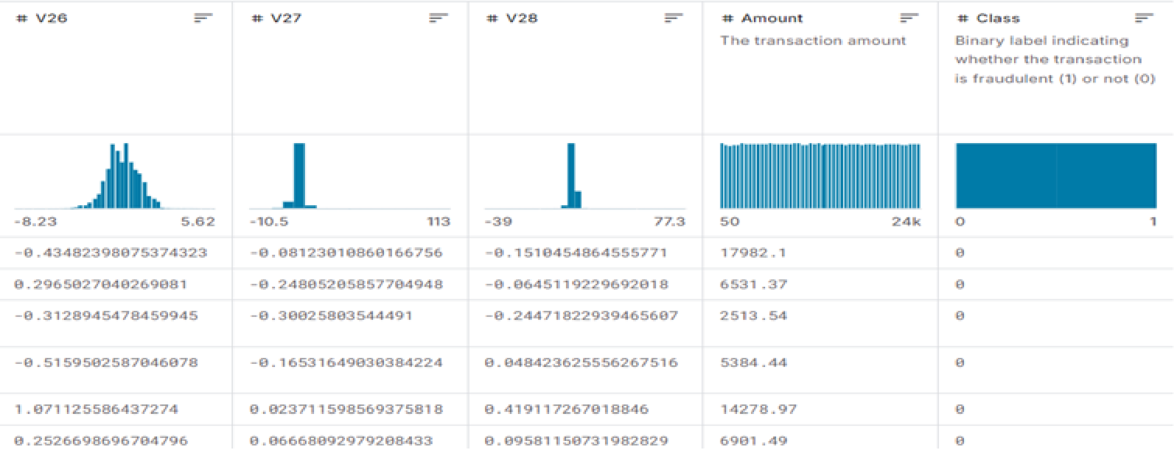
5.1. Selection of machine learning research datasets for predicting fraudulent anomalies in financial transactions

The first dataset contains 248,807 credit card transaction records, with 492 fraudulent (99:1 imbalance). It includes 28 anonymized features V0–V28, plus Time and Amount. The second is artificially generated and balanced (560,863 each class), structurally similar (V0–V28, no Time, removes uninformative ID). Used for training to avoid imbalance effects. The third, for real-world testing: 5,100 records (83:17), less anonymized but structurally different, so trained separately. Datasets referenced as 99:1, 50:50, and 83:17.



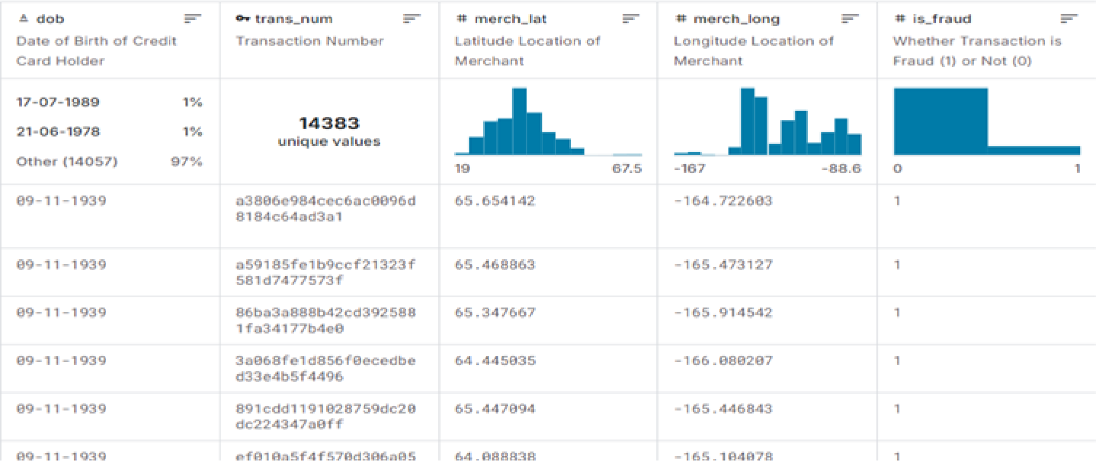
	# V26	# V27	# V28 abc	# Amount Transaction amount	# Class 1 for fraudulent transactions, 0 otherwise
7.52	-2.6 3.52	-22.6 31.6	-15.4 33.8	0 25.7k	0 1
3528	-0.189114843888824	0.133558376740387	-0.0210530534538215	149.62	0
8143	0.125894532368176	-0.00898309914322813	0.0147241691924927	2.69	0
35251	-0.139096571514147	-0.0553527940384261	-0.0597518405929204	378.66	0
2038	-0.221928844458407	0.0627228487293033	0.0614576285006353	123.5	0
19756	0.502292224181569	0.219422229513348	0.215153147499206	69.99	0
37034	0.105914779097957	0.253844224739337	0.0810802569229443	3.67	0
659	-0.257236845917139	0.0345074297438413	0.00516776890624916	4.99	0

Figure 3: A



# V26	# V27	# V28	# Amount The transaction amount	# Class Binary label indicating whether the transaction is fraudulent (1) or not (0)
-8.23 5.62	-10.5 113	-39 77.3	50 24k	0 1
-0.43482398075374323	-0.08123010860166756	-0.1510454864555771	17982.1	0
0.2965027040269081	-0.24805205857704948	-0.0645119229692018	6531.37	0
-0.3128945478459945	-0.30025803544491	-0.24471822939465607	2513.54	0
-0.5159502587046078	-0.16531649030384224	0.048423625556267516	5384.44	0
1.071125586437274	0.023711598569375818	0.419117267018846	14278.97	0
0.2526698696704796	0.06668092979208433	0.09581150731982829	6901.49	0

Figure 4: B



⚠ dob Date of Birth of Credit Card Holder	🗝 trans_num Transaction Number	# merch_lat Latitude Location of Merchant	# merch_long Longitude Location of Merchant	# is_fraud Whether Transaction is Fraud (1) or Not (0)
17-07-1989 1% 21-06-1978 1% Other (14057) 97%	14383 unique values	19 67.5	-167 -88.6	0 1
09-11-1939	a3806e984cec6ac0096d8184c64ad3a1	65.654142	-164.722603	1
09-11-1939	a59185fe1b9ccf21323f581d7477573f	65.468863	-165.473127	1
09-11-1939	86ba3a888b42cd3925881fa34177b4e0	65.347667	-165.914542	1
09-11-1939	3a068fe1d856f0ecedbed33e4b5f4496	64.445035	-166.080207	1
09-11-1939	891cdd1191028759dc20dc224347a0ff	65.447094	-165.446843	1
09-11-1939	ef010a5f4f570d306a05	64.088838	-165.104078	1

Figure 5: C

Figure 5 presented the three selected datasets for the study of fraudulent anomalies in financial transactions: (a) containing credit card transaction information over a specified period; (b) artificially generated with a balanced class distribution; (c) approximating real-world conditions.

5.2. Result and discussion

Research and testing ML models are crucial for building effective fraud detection systems in financial transactions. This validates methods in near-real conditions, where accuracy minimizes losses and fosters bank trust. Datasets with class imbalances (50:50, 83:17, 99:1) mirrored rare fraud scenarios, enabling full assessment of anomaly detection in highly skewed data. The study pinpointed optimal models for institutions that cut risks, boost efficiency, reduce false positives, lower security costs, and ensure live system stability. Below: detailed performance analysis per model across datasets. Logistic Regression model across the three datasets:

1. Dataset 50%/50%. The model demonstrated high accuracy at 96% with balanced precision and recall metrics, indicating its effectiveness on balanced data.
2. Dataset 99%/1%. Model accuracy sharply dropped to 10%, showing practically zero effectiveness in detecting fraudulent transactions (class 1) under strong class imbalance. Although the model correctly predicted all fraudulent transactions, it misclassified a significant number of legitimate operations as fraudulent. This may indicate potential overfitting, data leakage, or insufficient parameter tuning.
3. Dataset 83%/17%. Accuracy was 87%; however, the F1-score for class 1 significantly decreased (0.62), indicating frequent errors in fraud detection.

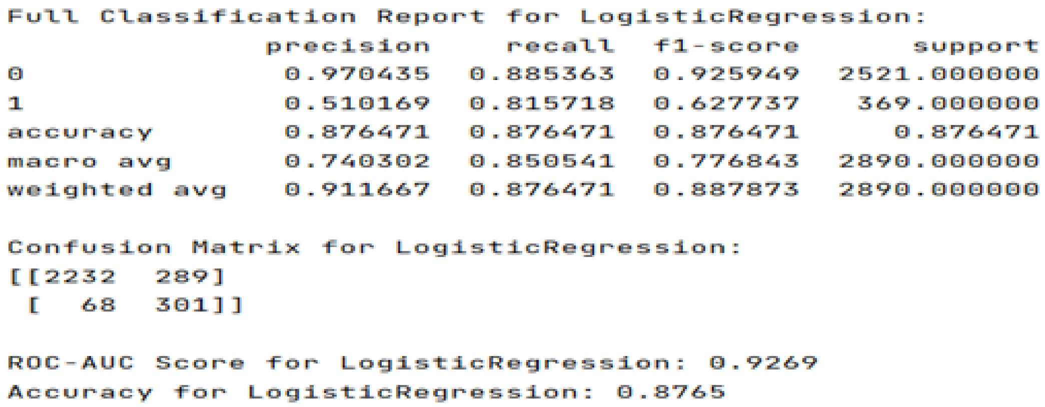
Table 1 and a screenshot (Fig. 4) below illustrate the accuracy metrics and training results of the Logistic Regression algorithm on the 83%/17% dataset, as mentioned in the previous sections.

Table 1
Logistic Regression of data 83%/17%

	Precision	Recall	F1-score	Support
0	0.97	0.88	0.92	2521
1	0.51	0.81	0.62	369

accuracy	0.87	0.87	0.87	0.87
Macro avg	0.74	0.85	0.77	2890
Weighted avg	0.91	0.87	0.88	2890

Figure 6 shows the LogisticRegression console output in the 87%/13% set with the corresponding metrics.



```
Full Classification Report for LogisticRegression:
              precision    recall  f1-score      support
0              0.970435  0.885363  0.925949  2521.000000
1              0.510169  0.815718  0.627737   369.000000
accuracy       0.876471  0.876471  0.876471     0.876471
macro avg      0.740302  0.850541  0.776843  2890.000000
weighted avg   0.911667  0.876471  0.887873  2890.000000

Confusion Matrix for LogisticRegression:
[[2232  289]
 [  68  301]]

ROC-AUC Score for LogisticRegression: 0.9269
Accuracy for LogisticRegression: 0.8765
```

Figure 6: Logistic Regression console output in the 87%/13% set

Decision Tree model:

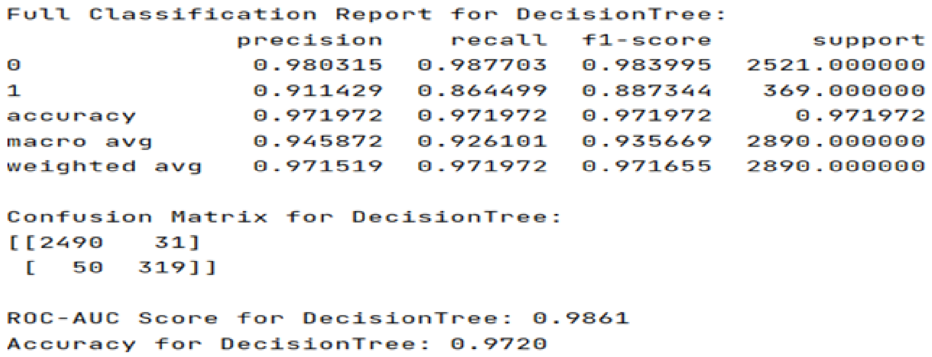
1. 50%/50% set. The model showed high precision and recall (96% for both classes) and an overall accuracy of 96%, indicating its good performance on uniform data.
2. 99%/1% set. The model maintained high accuracy for class 0, but precision for class 1 decreased significantly. The overall accuracy was 58%, indicating a problem with imbalance.
3. 83%/17% set. The overall accuracy of the model was high at 97%. The F1-score for class 1 was 0.88, which is better than Logistic Regression, but still shows a decrease in recall.

Below is Table 2 and a screenshot (Fig. 4) showing the accuracy metrics mentioned in the previous sections and the results of training the Decision Tree algorithm on the 83%/17% dataset.

Table 2
Logistic Regression of data 83%/17%

	Precision	Recall	F1-score	Support
0	0.98	0.98	0.98	2521
1	0.91	0.86	0.88	369
accuracy	0.97	0.97	0.97	0.97
Macro avg	0.94	0.92	0.93	2890
Weighted avg	0.97	0.97	0.98	2890

Figure 7 shows the Decision Tree console output in an 87%/13% set with corresponding metrics.



```
Full Classification Report for DecisionTree:
                precision    recall  f1-score      support
0                0.980315  0.987703  0.983995  2521.000000
1                0.911429  0.864499  0.887344   369.000000
accuracy         0.971972  0.971972  0.971972     0.971972
macro avg        0.945872  0.926101  0.935669  2890.000000
weighted avg     0.971519  0.971972  0.971655  2890.000000

Confusion Matrix for DecisionTree:
[[2490   31]
 [  50  319]]

ROC-AUC Score for DecisionTree: 0.9861
Accuracy for DecisionTree: 0.9720
```

Figure 7: Decision Tree console output in the 87%/13% set

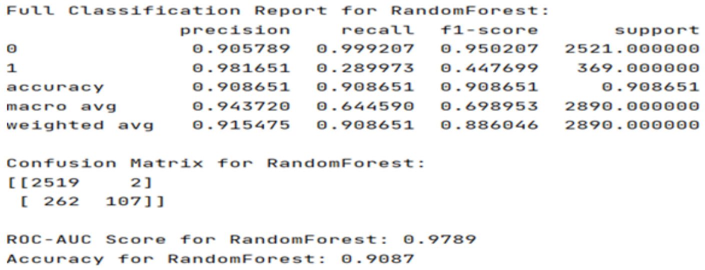
Random Forest model: 50%/50% set: Demonstrated impressive results: 98% accuracy and high F1-score values for both classes. 99%/1% set: Despite 71% accuracy, precision for class 1 was close to 0, indicating a lack of recognition of fraudulent transactions. 83%/17% set: Accuracy remained at 90%, but the F1-score for class 1 dropped to 0.44, indicating problems with detecting the smaller class.

Below is Table 3 and a screenshot (Fig. 8) showing the accuracy metrics mentioned in the previous sections and the results of training the Random Forest algorithm on the 83%/17% dataset.

Table 3
Random Forest of data 83%/17%

	Precision	Recall	F1-score	Support
0	0.90	0.99	0.95	2521
1	0.90	0.28	0.44	369
accuracy	0.90	0.90	0.90	0.90
Macro avg	0.94	0.64	0.69	2890
Weighted avg	0.91	0.90	0.88	2890

Figure 8 shows the Random Forest console output in an 87%/13% set with corresponding metrics.



```
Full Classification Report for RandomForest:
                precision    recall  f1-score      support
0                0.905789  0.999207  0.950207  2521.000000
1                0.981651  0.289973  0.447699   369.000000
accuracy         0.908651  0.908651  0.908651     0.908651
macro avg        0.943720  0.644590  0.698953  2890.000000
weighted avg     0.915475  0.908651  0.886046  2890.000000

Confusion Matrix for RandomForest:
[[2519    2]
 [ 262  107]]

ROC-AUC Score for RandomForest: 0.9789
Accuracy for RandomForest: 0.9087
```

Figure 8: Random Forest console output in the 87%/13% set

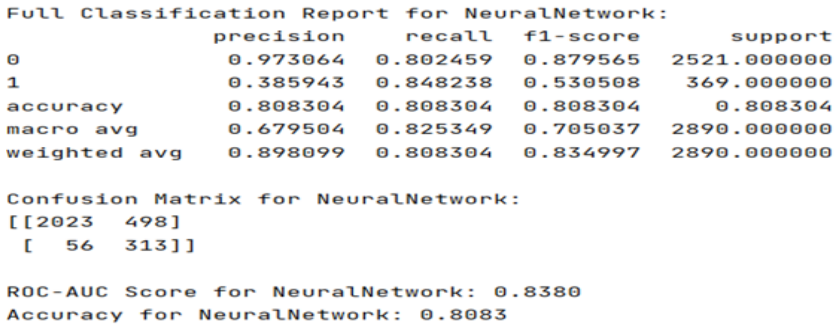
Neural Network (MLP Classifier) model:50%/50% set: Showed almost perfect performance with 98% accuracy and an F1-score of 0.99 for both classes. 99%/1% set. There was a significant problem with precision for class 1, although the overall accuracy of 83% remained acceptable. 83%/17%. Accuracy dropped to 80%, and the F1-score for class 1 was only 0.53, indicating insufficient adaptation to the imbalance. Below is Table 4 and a screenshot (Fig. 6) showing the accuracy metrics mentioned in the previous sections and the results of training the MLP Classifier algorithm on the 83%/17% dataset.

Table 4

MPL Classifier of data 83%/17%

	Precision	Recall	F1-score	Support
0	0.97	0.80	0.87	2521
1	0.38	0.84	0.53	369
accuracy	0.80	0.80	0.80	0.90
Macro avg	0.67	0.82	0.70	2890
Weighted avg	0.89	0.90	0.83	2890

Figure 9 shows the Neural Network console output in the 87%/13% set with the corresponding metrics.



```
Full Classification Report for NeuralNetwork:
              precision    recall  f1-score      support
0              0.973064  0.802459  0.879565  2521.000000
1              0.385943  0.848238  0.530508   369.000000
accuracy       0.808304  0.808304  0.808304     0.808304
macro avg      0.679504  0.825349  0.705037  2890.000000
weighted avg   0.898099  0.808304  0.834997  2890.000000

Confusion Matrix for NeuralNetwork:
[[2023  498]
 [  56  313]]

ROC-AUC Score for NeuralNetwork: 0.8380
Accuracy for NeuralNetwork: 0.8083
```

Figure 9: MPL Classifier console output in the 87%/13% set

Gradient Boosting model:50%/50% set. The experiment demonstrated high efficiency on a uniform data set with an accuracy of 97%. 99%/1% set. The overall accuracy was 47% due to high imbalance, which led to a significant decrease in the F1-score for class 1. 83%/17% set. The accuracy reached 99%. The model handled the imbalanced data well, although the F1-score for class 1 decreased slightly to 0.97.

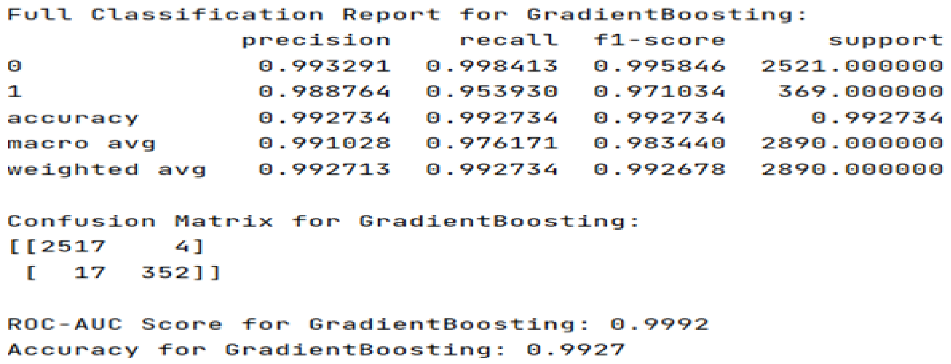
Below is Table 5 and a screenshot (Fig. 7) showing the accuracy metrics mentioned in the previous sections and the results of training the Gradient Boosting algorithm on the 83%/17% dataset.

Table 5

GradientBoosting of data result 83%/17%

	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	2521
1	0.98	0.95	0.97	369
accuracy	0.99	0.99	0.99	0.90
Macro avg	0.99	0.99	0.98	2890
Weighted avg	0.99	0.9	0.99	2890

Figure 10 shows the Gradient Boosting console output in an 87%/13% set with corresponding metrics.



```
Full Classification Report for GradientBoosting:
              precision    recall  f1-score      support
0              0.993291  0.998413  0.995846  2521.000000
1              0.988764  0.953930  0.971034   369.000000
accuracy       0.992734  0.992734  0.992734     0.992734
macro avg      0.991028  0.976171  0.983440  2890.000000
weighted avg   0.992713  0.992734  0.992678  2890.000000

Confusion Matrix for GradientBoosting:
[[2517    4]
 [  17  352]]

ROC-AUC Score for GradientBoosting: 0.9992
Accuracy for GradientBoosting: 0.9927
```

Figure 10: Gradient Boosting console output in the 87%/13% set

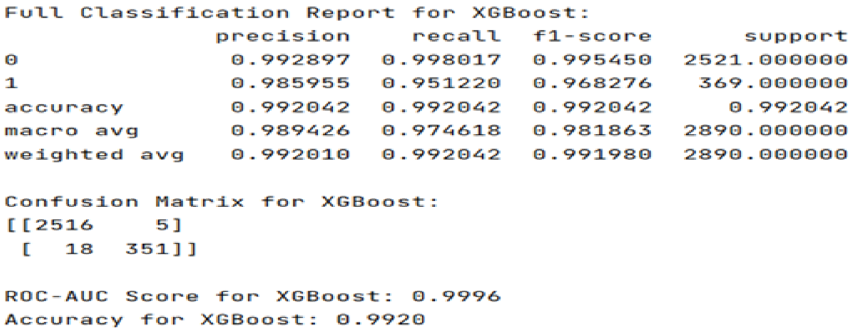
XGBoost model: 50%/50% set: The model showed a high accuracy of 97% with well-balanced metrics for both classes. 99%/1% set. The model demonstrated the best result among all models for the 99%/1% set, achieving 99% accuracy and an F1-score for class 1 of 0.96. 83%/17% set. The model maintained high accuracy of 99%, adapting well to partially imbalanced data.

Below is Table 6 and a screenshot (Fig. 8) showing the accuracy metrics mentioned in the previous sections and the results of training the XGBoost algorithm on the 83%/17% dataset.

Table 6
XGBoost of data result 83%/17%

	Precision	Recall	F1-score	Support
0	0.99	0.99	0.99	2521
1	0.98	0.95	0.96	369
accuracy	0.99	0.99	0.99	0.99
Macro avg	0.98	0.97	0.98	2890
Weighted avg	0.99	0.99	0.99	2890

Figure 11 shows the XGBoost console output in an 87%/13% split with the corresponding metrics.



```
Full Classification Report for XGBoost:
              precision    recall  f1-score      support
0              0.992897  0.998017  0.995450  2521.000000
1              0.985955  0.951220  0.968276   369.000000
accuracy       0.992042  0.992042  0.992042     0.992042
macro avg      0.989426  0.974618  0.981863  2890.000000
weighted avg   0.992010  0.992042  0.991980  2890.000000

Confusion Matrix for XGBoost:
[[2516    5]
 [  18  351]]

ROC-AUC Score for XGBoost: 0.9996
Accuracy for XGBoost: 0.9920
```

Figure 11: XGBoost console output in the 87%/13% set

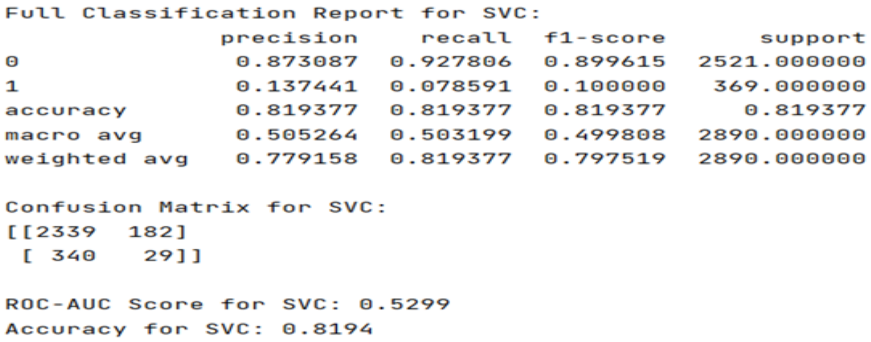
SVC model: 50%/50% set. The model showed average performance with an accuracy of 52%, indicating its low ability to detect both classes. 99%/1% set. The model shows a sharp drop in the F1-score for class 1, with an accuracy of only 55%. 83%/17% set. The model's accuracy was 81%, but the F1-score for class 1 was only 0.10, indicating poor recognition of the smaller class.

Below is Table 7 and a screenshot (Fig. 9) showing the accuracy metrics mentioned in the previous sections and the results of training the SVC algorithm on the 83%/17% dataset.

Table 7
SVC of data result 83%/17%

	Precision	Recall	F1-score	Support
0	0.87	0.92	0.89	2521
1	0.13	0.87	0.1	369
accuracy	0.81	0.81	0.81	0.99
Macro avg	0.50	0.50	0.49	2890
Weighted avg	0.77	0.81	0.79	2890

Figure 12 shows the SVC console output in an 87%/13% set with corresponding metrics.



```
Full Classification Report for SVC:
              precision    recall  f1-score      support
0              0.873087  0.927806  0.899615  2521.000000
1              0.137441  0.078591  0.100000   369.000000
accuracy       0.819377  0.819377  0.819377     0.819377
macro avg      0.505264  0.503199  0.499808  2890.000000
weighted avg   0.779158  0.819377  0.797519  2890.000000

Confusion Matrix for SVC:
[[2339  182]
 [ 340   29]]

ROC-AUC Score for SVC: 0.5299
Accuracy for SVC: 0.8194
```

Figure 12: SVC console output in the 87%/13% set

The same studies were also conducted on two other datasets. The results of these studies will be presented in a table showing the accuracy and ROC-AUC score for each algorithm used. Below is a table with the results of the algorithms in the 50%/50% dataset.

Table 8
Research of data result 83%/17%

Algorithms	ROC-AUC	Accuracy
Logistic Regression	0.9931	0.9606
Decision Tree	0.9833	0.9601
Random Forest	0.9993	0.9839
MLP Classifier	1.0000	0.9998
Gradient Boosting	0.9983	0.9793
XGBoost	0.9985	0.9772
SVC	0.7396	0.5214

The following table shows the results of studies in a 99%/1% set.

Table 9
Research of data result 99%/1%

Algorithms	ROC-AUC	Accuracy
Logistic Regression	0.9815	0.1066

Decision Tree	0.8326	0.5820
Random Forest	0.9641	0.7169
MLP Classifier	0.9840	0.8322
Gradient Boosting	0.9720	0.4734
XGBoost	0.9748	0.4821
SVC	0.0644	0.5512

In general, the results of the study allow us to draw the following conclusions about the effectiveness of the models used. The XGBoost and Gradient Boosting algorithms demonstrated the highest performance for all class imbalance options, indicating their high adaptability. In contrast, Logistic Regression and Support Vector Classifier (SVC) proved ineffective in cases of significant imbalance, losing their ability to accurately classify fraudulent transactions. Methods based on tree structures generally cope better with partially unbalanced data, but require careful tuning when working with heavily skewed samples. Overall, most of the tested models confirmed their reputation in the context of detecting fraudulent anomalies.

6. Conclusions

The article examines the application of machine learning for predicting fraudulent anomalies in financial transactions. The aim of the study was to develop an effective framework capable of identifying suspicious transactions at early stages, thereby minimizing financial losses, reputational risks, and legal costs. According to the Association of Certified Fraud Examiners (ACFE), annual losses from financial fraud have exceeded 5% of organizational revenue worldwide, and the average loss in 2024 increased for the first time since 2016.

Traditional fraud detection methods, which rely on static rules and manual analysis, are no longer adequate to meet modern challenges due to the growing volume of data and the increasing complexity of financial operations. These methods cannot process large datasets in a timely manner or recognize complex anomalous behavior patterns. Consequently, the implementation of machine learning and artificial intelligence technologies has become increasingly relevant, as they enable the creation of automated, adaptive systems for real-time fraud prediction.

The study developed a step-by-step approach for detecting fraudulent anomalies using machine learning, which includes data preparation and preprocessing, building and training classification models, and evaluating their effectiveness. The performance of seven popular machine learning algorithms was analyzed, including linear regression, decision trees, random forest, neural networks, Gradient Boosting, XGBoost, and SVC.

The research demonstrated that ensemble methods such as Random Forest, Gradient Boosting, and XGBoost are the most suitable for fraud detection tasks. These models consistently delivered high performance across different datasets, including when working with imbalanced data, which is typical of real financial transactions. The study also confirmed the importance of proper parameter tuning and accounting for class imbalance to improve model accuracy.

Applying the proposed approach enables financial institutions to significantly enhance operational efficiency, minimize financial losses, and strengthen client trust, providing comprehensive protection for the financial system in a dynamic digital environment.

Acknowledgements

Thanks our colleagues for their valuable advice and constructive feedback, which contributed to refining the methodology and interpreting the results.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] E. W. T. Ngai, Y. Hu, Y. H. Wong, Y. Chen, X. Sun, Data mining techniques in financial fraud detection: classification framework, Decision Support Systems 50 (2011) 559–569.

[2] Y. Zhou, K. Shu, H. Liu, Detecting fraud in online transactions using machine learning: a review, ACM Computing Surveys 54 (2021) 1–36.

[3] S. Bhattacharyya, S. Jha, K. Tharakunnel, J. C. Westland, Data mining for credit card fraud: a comparative study, Decision Support Systems 50 (2011) 602–613.

[4] J. Jurgovsky, G. Granitzer, K. Ziegler, S. Calabretto, P. Portier, Sequence classification for credit-card fraud detection, Expert Systems with Applications 100 (2018) 234–245.

[5] C. Whitrow, D. J. Hand, P. Juszczak, D. Weston, N. M. Adams, Transaction aggregation for credit card fraud detection, Data Mining and Knowledge Discovery 18 (2009) 30–55.

[6] Y. Chen, X. Xu, J. Liu, J. Hu, Blockchain-based financial fraud detection: a review, IEEE Access 8 (2020) 111697–111707.

[7] D. Uhryn, Y. Ushenko, V. Lytvyn, Z. Hu, O. Lozynska, V. Ilin, A. Hostiuk, Intelligent GIS model for migration forecasting, Int. Journal of Modern Education and Computer Science 15 (2023) 69–79. https://doi.org/10.5815/ijmecs.2023.04.06

[8] Association of Certified Fraud Examiners, Report to the Nations: 2022 Global Study on Occupational Fraud and Abuse, ACFE, Austin, 2022. URL: https://www.acfe.com/report-to-the-nations/2022/

[9] PricewaterhouseCoopers, Global Economic Crime and Fraud Survey 2023, PwC, 2023. URL: https://www.pwc.com/gx/en/services/forensics/economic-crime-survey.html

[10] International Monetary Fund, Fraud in financial institutions, 2025. URL: https://www.imf.org/external/index.htm

[11] Scikit-learn, Machine learning in Python – documentation, 2025. URL: https://scikit-learn.org/stable/documentation.html

[12] XGBoost, Extreme Gradient Boosting documentation, 2025. URL: https://xgboost.readthedocs.io/

[13] Google AI Blog, Fighting fraud with machine learning, 2025. URL: https://ai.googleblog.com/

[14] Kaggle, Fraud detection datasets, 2025. URL: https://www.kaggle.com/

[15] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006. URL: https://www.springer.com/gp/book/9780387310732

[16] Ministry of Finance of Ukraine, Financial security, 2025. URL: https://mof.gov.ua/en/

[17] Financial Times, Combating financial fraud with AI, 2025. URL: https://www.ft.com/

[18] International Journal of Financial Studies, AI in fraud detection, 2025. URL: https://www.mdpi.com/journal/ijfs

[19] S. Vladov, V. Vysotska, V. Sokurenko, O. Muzychuk, L. Chyrun, The Intelligent Data Measurement System Using Neural Network Technologies and Fuzzy Logic Under Operating Implementation Conditions, Big Data and Cognitive Computing 8:12 (2024) 189. https://doi.org/10.3390/bdcc8120189

[20] Visa Inc., Fraud prevention with advanced analytics, 2025. URL: https://usa.visa.com

[21] D. Uhryn, V. Andrunyk, L. Chyrun, N. Antonyuk, I. Dyyak, O. Naum, Service-oriented architecture as integration platform in tourism, in: Proc. 2nd Int. Workshop on Modern Machine Learning Technologies and Data Science (MoMLeT+DS 2020), CEUR-WS 2631, 2020, 221–236. URL: https://ceur-ws.org/Vol-2631/paper17.pdf