European Journal of Interdisciplinary Research and Development
Volume-21 November 2023
Website: www.ejird.journalspark.org ISSN (E): 2720-5746

ENHANCING FRAUD DETECTION IN FINANCIAL TRANSACTIONS: A COMPARATIVE ANALYSIS OF MACHINE LEARNING ALGORITHMS

Duraid Muneer Mohammd 1,
Kassem Mohamed Danach2
1-Department of Computer and Telecom, Faculty of Engineering, Islamic University of Lebanon, Beirut, Lebanon; driedsara2@gmail.com
2-Department of Information Technology and Management Systems, Faculty of Business Administration, Al Maaref University; Kassem.danach@mu.edu.lb

Abstract
In the modern financial landscape, fraudulent personal information in financial transactions threatens data integrity and security. There is potential for machine learning to identify and mitigate fraud threats. This comparative study examines machine learning techniques for detecting and preventing the use of fraudulent personal information in financial transactions. Using a large dataset of financial transaction records, logistic regression, k-nearest neighbors, decision trees, gradient boosting, and ensemble methods such as Adaboost, XGboost, LightGBM, and CatBoost are evaluated. Presented is a comprehensive analysis and comparison of each algorithm's ability to detect and eradicate false personal information. The findings highlight the significance of using the optimal algorithm to optimize the detection and prevention of financial transaction fraud. The study also sheds light on the interpretability and scalability of the algorithms, allowing for the incorporation of robust and flexible machine learning approaches in financial security and data protection. In conclusion, this study offers crucial insights regarding the selection and application of machine learning algorithms to combat the use of fraudulent personal information in financial transactions. The findings highlight the need for sophisticated machine learning solutions to combat financial crime and strengthen data integrity and security standards.

Keywords: machine learning, Financial Transactions, logistic regression, k-nearest neighbors, decision trees, gradient boosting, Adaboost, XGboost, Light GBM, and Cat Boost.

Introduction

Financial transactions are threatened by fraudulently obtained personal data in worldwide banking systems. Effective procedures to detect and eliminate identity theft and credential fraud are needed due to rising rates[1]. Due to these challenges, machine learning is a promising method for financial transaction verification and validation[2]. Using machine learning, financial transactions can be protected from fake personal information. Advanced algorithms detect anomalies, patterns, and disparities that may suggest fraud. Anomaly detection algorithms reveal data trends or variations that indicate fraud, such as using fake personal information[3]. Supervised learning methods that use personal data attributes help classify transactions as real or fraudulent[4]. Clustering, an unsupervised learning method, helps categorize transactions based on personal information by discovering clusters that differ from the norm, which may

suggest fraud[5]. Ensemble learning uses many models to increase forecast accuracy and generalizability, detecting fake personal data by leveraging varied methods[6]. Deep learning algorithms like neural networks can detect sophisticated fraud using complex data patterns and connections related to fraudulent personal information[4]. NLP may identify language discrepancies in financial transaction textual data that may indicate fraud. Feature engineering also helps machine-learning models detect fraudulent information by selecting and transforming important personal information data features. By lowering data dimensionality while maintaining vital information, PCA and t-SNE help identify faked personal information. Understanding how machine learning algorithms make decisions helps increase the accuracy and transparency of spotting fake personal data[6]. When seamlessly integrated, these machine-learning methods can predict and uncover personal data fraud in banking transactions, improving financial system security. The study focuses on how machine learning can detect and prevent counterfeit personal information fraud in worldwide financial transactions. The paper begins by listing the most effective ways to detect and manage personal data fraud. It also examines machine learning's potential to improve banking transaction security and validation. The discussion then covers supervised and unsupervised learning, anomaly detection, ensemble learning, deep learning, natural language processing, and feature engineering. Applying many approaches on the same data to compare efficiency. The research describes how these methods might be used to spot abnormal patterns and anomalies that indicate counterfeit personal information fraud. Machine learning models' transparent decision-making procedures improve trust and reliability in identifying fake personal data. Finally, a separate section compares all cases' efficiency and accuracy. The rest of the study structured in the following way: the second section introduces Machine-learning tools to get rid of the Fake Personal Information in Financial Transactions. The third section begins with a review of earlier work on a few of the most relevant algorithms and applications. The fourth section show the proposed work and the fifth provides a results and discussion for the proposed work. Lastly, section six provides a conclusion of this paper.

Literature review

In this section, numerous studies have discussed fraudulent transactions. In [7], Rapid global technological improvements are driving daily card use above cash, according to the report. MasterCard became a popular online buying card, but damage and fraud incidents increased. These illicit transactions threatened financial stability, making detection essential. The article employs deep learning, machine learning, and models like Bidirectional Long Short-Term Memory and Bidirectional Gated Recurrent Unit to differentiate lawful from fraudulent transactions. Their model outperformed machine learning classifiers with 91.37% accuracy.

S, Varun, [8]. The researcher wants to use machine learning and neural networks to predict fraudulent and non-fraudulent transactions. The project uses classification machine learning algorithms, statistical methods, calculus (including differentiation and the chain rule), and linear algebra to build complex machine learning models to understand the dataset and predict fraudulent and non-fraudulent transactions based on transaction time and amount. They achieved 94.84% accuracy using logistic regression, 91.62% with naive Bayes, and 92.88% with decision

trees. In deep learning, they used an Artificial Neural Network (ANN) that surpassed all other methods with 98.69% accuracy.

Mehbodniya et al. [9]. This study examines machine learning and deep learning methods for credit card fraud detection. Naive Bayes, Logistic Regression, K-Nearest Neighbor (KNN), Random Forest, and Sequential Convolutional Neural Network train models using regular and irregular transaction variables to detect credit card fraud. The model's accuracy is assessed using public data. Naive Bayes, Logistic Regression, KNN, Random Forest, and Sequential Convolutional Neural Networks have accuracy scores of 96.1%, 94.8%, 95.89%, 97.58%, and 92.3%. Comparative investigation shows that KNN is more accurate than other approaches.

In [10], the authors concentrated on addressing the issue of class imbalance in fraud detection using machine learning algorithms. They also highlighted the summarized results and weaknesses observed when using labeled credit card fraud datasets. Their conclusion emphasized the ineffectiveness of imbalanced classification when dealing with highly skewed data, noting that existing methods were expensive and prone to false alarms.

In [11], Oversampling is necessary for DT, LR, and RF algorithms, which are used to detect fraudulent use of credit cards, because the dataset is unbalanced. Following the usage of oversampling, the dataset was composed of 60% legitimate transactions and 40% fraudulent ones, and the R programming language was utilized for the implementation. Along with the measurements of sensitivity, error rates, and specificity, the accuracy rates came in at 90.0% LR, 95.5% RF, and 94.3% DT respectively. RF performed the best when compared to these other algorithms.

In [12], The use of RF, SVM, and LR allowed for the identification of fraudulent activity using both automatic and guided classification techniques. The purpose of the research was to create a model for rating risks, and it was discovered that RF performed very well and achieved the best accuracy. This algorithm's real-world usefulness was demonstrated by the fact that it was simple to construct yet successful even when applied to big datasets.

In [13], several ML algorithms to assess performance on an imbalanced dataset, testing SVM, RF, DT, and LR on pre-processed and raw data. Accuracy rates were SVM 97.5%, RF 98.6%, DT 95.5%, and LR 97.7%. RF excelled with large datasets, but its speed was a limitation. For highly pre-processed data, SVM emerged as a viable alternative.

Kousika et al. [14]. This research aims to identify instances of financial fraud in business transactions through the utilization of machine algorithms. The paper introduces an algorithm rooted in Machine Learning for detecting credit card fraud, addressing the problem of fraudulent transactions. This framework significantly enhances the ability to detect fraudulent card activity exponentially. The outcomes demonstrate that the accuracy rates for Random Forest, Support Vector Machine, and KNN classifiers are 94.84%, and 89.46%, and, notably, Random Forest exhibits swift detection of new fraud cases.

In [15], employed SVM to distinguish between valid and fraudulent transactions by analyzing cardholders' past transaction behaviors. Any new transaction that deviated from this pattern was marked as fraudulent, achieving a fraud detection score of 91% with SVM.

In [16], proposed a deep neural network approach for fraud detection. They employed log transformation to address data skew issues and focal reduction for training on challenging

examples. The results demonstrated superior performance compared to classical models like SVM and LR.

Abdulsattar et al. [17]. This research presents a binary classification problem involving identifying fraudulent or legitimate transactions using five machine learning algorithms. The accuracy percentages for Task1 and Task2 datasets were within the range of 97.78% to 98.1%, with no significant difference between them. The RF classifier showed the highest kappa statistic value, while SGD and J48 classifiers had the lowest values. The SGD classifier had the least favorable results in Task 2, while RF outperformed other classifiers in terms of Kappa statistics and MCC values. In conclusion, these classifiers demonstrated comparable performance across both datasets.

In [18], introduced a hybrid approach using DT and Rough Set methods for credit card fraud detection. They utilized WEKA and MATLAB software for their work and found that their proposed technique performed well with an accuracy rate of 84.25% after ten executions.

[19], introduced the Lightgbm algorithm for fraud detection and compared it with Logistic Regression, SVM, and Xgboost. Lightgbm achieved an accuracy rate of 98%, outperforming Logistic Regression (92.60%), SVM (95.20%), and Xgboost (97.10%).

In [20], the REDBSCAN algorithm reduces the number of samples and maintains data integrity. A comparison with the SVM technique revealed that SVDD achieved an AUC of 97.75%, while SVM achieved 94.60%. When SVDD was combined with REDBSCAN, the processing time was reduced significantly to 1.69 seconds, making REDBSCAN a faster and more desirable option.

Nami et al. [21] developed a dynamic Random Forest and KNN system for payment card fraud detection. Initially, they extracted dataset attributes to understand cardholder behavior. A created similarity metric was used to compare new and current cardholders. They used KNN to train a Dynamic Random Forest model to predict transaction outcomes cost-effectively.

Carta et al. [22] The "Prudential Multiple Consensus" model combines classifiers like Naïve Bayes, Multilayer Perceptron, Adaptive Boosting, Gradient Boosting, and Random Forest. First, the model determines whether a transaction is genuine if the majority of classifiers classify it as such with a probability greater than the average probability of all classifiers. After all classifiers are conducted in the first stage, the classifier with the greatest vote decides in the second step.

Mints et al. [23]. This research aims to develop efficient models for identifying fraudulent activities in digital payment systems using automated machine learning and Big Data analysis algorithms. The authors propose methods to improve the information repository for spotting fraudulent transactions and justify performance metrics for constructing and comparing models. The proposed algorithms achieve a classification quality of 0.977-0.982, surpassing traditional classifiers and reducing model synthesis time. The models successfully identify up to 85.7% of fraudulent transactions, with high fraud detection accuracy ranging from 79-85%. Implementing these findings could reduce financial and temporal investments in anti-fraud systems and enhance financial transaction monitoring.

Proposed Process

This section describes the proposed system's design and architecture, including its core components and functions to handle financial transaction fraud detection concerns as it shown in figure (1)

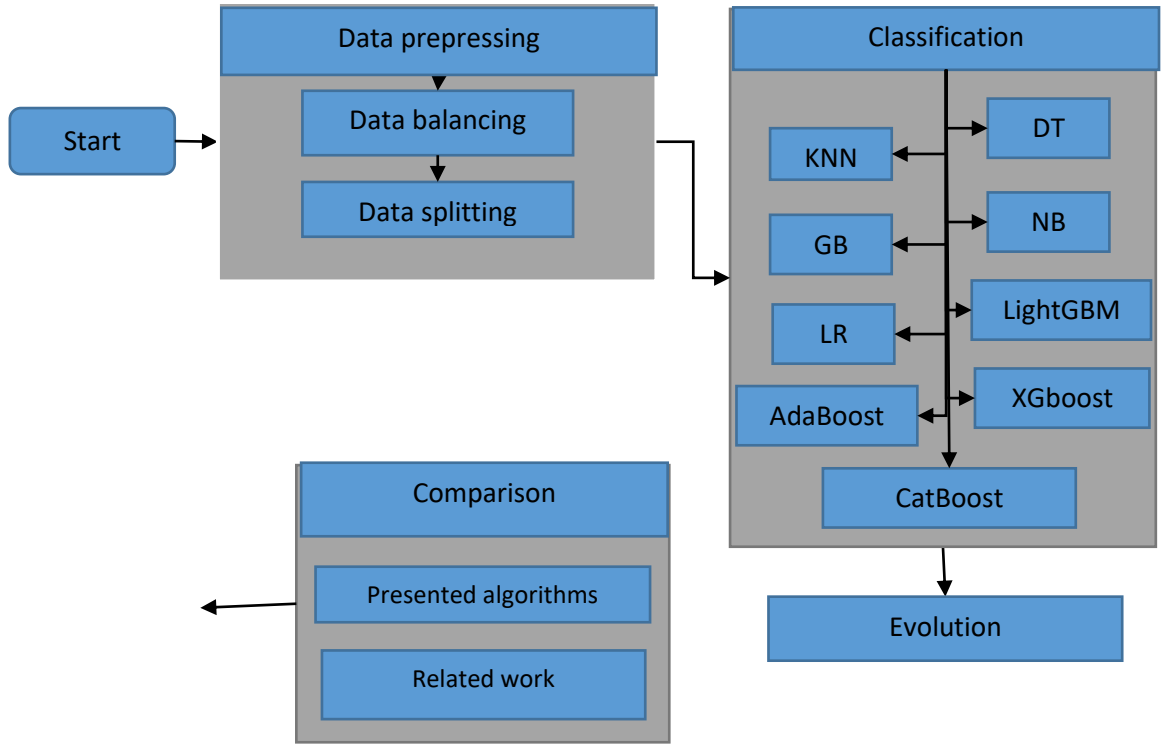

Figure (1) the proposed system flowchart

Dataset used in presented work

This study used Kaggle data on "fraudulent transaction detection". The dataset includes 1.75 million simulated user transactions from January to June 2023 across terminals.

Machin learning algorithms

Several machine learning methods were selected and subsequently applied to a common dataset in order to evaluate their performance and derive findings that could be of value to academics for comparative purposes

K-Nearest Neighbor Algorithm

Is a simple classification algorithm for machine learning, consisting of a training sample with a vector and class label. The algorithm stores feature vectors and class labels, identifying the test sample's class. The distance to each sample is computed, and the ideal class is determined by the majority vote of its neighbors. [24]. The ideal choice of k is determined by the dataset. k is defined as k = sqrt (N)/2, where N is the number of training samples. However, the best method is to try multiple K values to see which one delivers the best results. The KNN algorithm employs a wide range of distance measurements.

Decision tree

Decision Trees (DT) are employed for classifying instances by arranging them based on their feature values. Within a decision tree, every node symbolizes a feature within an instance requiring classification, and each branch signifies a potential value for that feature. Commencing from the root node, instances are organized and categorized according to their feature values. Decision trees serve as predictive models in decision tree learning, a technique employed in data mining and machine learning, where insights regarding an object are translated into predictions regarding its target value. More fitting names for these tree models can be "Classification trees" or "Regression trees [25].

Gradient Boosting

Gradient Descent is a machine learning technique used for first-order iterative optimization to find the local minimum of a function by taking steps proportional to the negative gradient at the current point. Gradient ascent is an optimization technique that involves taking steps proportional to the positive gradient to reach a local maximum of a function [26]. It determines parameter values (coefficients) to minimize a cost function, which assesses the difference between predicted and actual values. The algorithm adjusts coefficients until convergent [27].

Gradient Boosting's fundamental equation appears as:

$$F(x) = \Sigma m=1M\ fm(x) \quad (1)$$

Where:

F(x) is the final prediction of the model for a given input x.

M is the total number of learners in the model.

FM (x) is the prediction of the m-th learner for the input x. Residues from previous learners are used to calculate residual errors

Naive Bayes

It is a classification method that is predicated on the idea of feature independence and is based on the ideas of Bayes' Theorem. A Naive Bayes classifier, in its simplest form, functions on the premise that the existence of one feature in a class is independent of the existence of any other feature. Naive Bayes is predominantly employed in the field of text categorization, primarily for tasks involving classification and clustering that rely on conditional probability calculations [28]. This technique tells us how the other variables influence the probability of an event.

$$P(Y/X_1, X_2, \dots, X_n) = \frac{P(X1,X2,\dots,Xn/Y)P(Y)}{P(X1,X2,\dots,Xn)} \quad (2)$$

AdaBoost

The boosting approach is a learning technique that assigns positive integer weights to instances. It modifies the weights of instances connected to the classifier's output to construct a classifier that aids in the precise classification of complex cases. These instance weights are adjusted based on the outcomes of the newly created classifier. AdaBoost serves as an indicator of the compatibility of combined classifiers with the data, enabling us to identify experts who can collaborate effectively [29].

XGBoost

A gradient-boosting decision tree ensemble that scales well. Like gradient boosting, XGBoost maximizes a loss function to add to the objective function. XGBoost only uses decision trees as basis classifiers, therefore a loss function variation adjusts tree complexity[30].

$$L_{xgb} = \sum_{i=1}^{N} L\left(y_i, F(\mathbf{x}_i)\right) + \sum_{m=1}^{M} \Omega(h_m) \quad (3)$$

$$\Omega(h) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2 \quad (4)$$

Where T is the number of leaves on the tree and w is the leaf output scores. This loss function can be used in decision trees' split criterion to pre-prune. Higher values simplify trees. The value defines the minimum loss reduction gain needed to separate internal nodes. A regularization parameter in XGBoost called shrinkage reduces the additive expansion step size. Finally, using tree depth, etc. can limit tree complexity. Models train faster and use less storage when tree complexity is reduced.

LightGBM

Is a gradient-boosting framework that has been popular in the machine-learning community and is especially effective at handling challenging regression and classification issues. It is open-source software that Microsoft created. Microsoft has introduced a groundbreaking GBDT architecture for machine learning tasks, offering remarkable accuracy and efficiency. This scalable approach uses leaf-wise splitting, boosting speed, and reducing loss, but also introduces complexity and the risk of overfitting, unlike traditional boosting algorithms. [31].

Overfitting can be avoided by specifying the depth to which splitting will occur. GOSS and EFB approaches are used by LightGBM to minimize sampled data and feature size [32]. The difficulty of histogram construction is reduced from (O (data * feature)) to (O (data2 * bundles)), where data2 is data and bundles is a feature [31]. As seen below, the LightGBM method minimizes the anticipated value of a loss function L, (y, f (x)):

$$\hat{f}=\min E_{y,x} L(y,f(x)) \quad (5)$$

CatBoost

CatBoost stands as a machine learning approach employed in supervised learning scenarios, with a specific emphasis on classification and regression tasks. It derives its name from "Categorical Boosting" and is designed to effectively manage categorical attributes, rendering it highly applicable across various real-world use cases. CatBoost is a gradient boosting method that marries the strengths of gradient boosting with a distinct focus on handling categorical features. It was developed by Yandex, a prominent Russian multinational IT company, with the primary aim of resolving challenges associated with categorical data, including issues like high cardinality, missing values, and the need for extensive preprocessing [33].

Performance Evaluation

This study's evaluation metrics for classification performance are precision, accuracy, F1-score, and recall.

Accuracy (ACC):

Correct predictions divided by total cases investigated is the accuracy metric. It can be calculated using this equation:

$Accuracy = (TP + TN)/(TP + TN + FP + FN)$ (6)

Precision

Compare the percentage of correct detections to the total positive detections. It can be calculated using (7).

$Precision = TP/(TP + FP)$ (7)

Recall/Sensitivity

Can computed by through dividing total number of the correct positive predictions by total number of the positive cases. The Equation for calculating this metric is as follows[34]:

$Recall = TP/(TP + FN)$ (8)

F1-score

The weighted harmonic mean, calculated by recall and precision

$F1 = 2 * (Specificity * Recall)/(Specificity + Recall)$(9)

"TP" (true positives) and "TN" (true negatives) show positive and negative images identified correctly by classifier. False positives (FP) are positive photos misclassified as negatives, while false negatives (FN) are negative images misclassified as positives[35], [36].

Results and Discussion

In this section the important result of the study work is shown

5.1 preprocessing phase

Our initial research required extensive data cleaning to assure the financial transaction dataset's integrity and trustworthiness. The imported dataset was meticulously cleaned, removing empty columns, superfluous fields, and missing information. This crucial phase prepared the information for modeling because correct and complete data is necessary for any analytical activity.

After cleaning, we used an oversampling approach to address financial transaction data's class imbalance. Class imbalance can dramatically affect machine learning model performance, especially in fraud detection. The oversampling approach was used to balance the dataset by increasing minority situations, such as fraudulent transactions.

Figure (2) shows how oversampling changed data distribution.

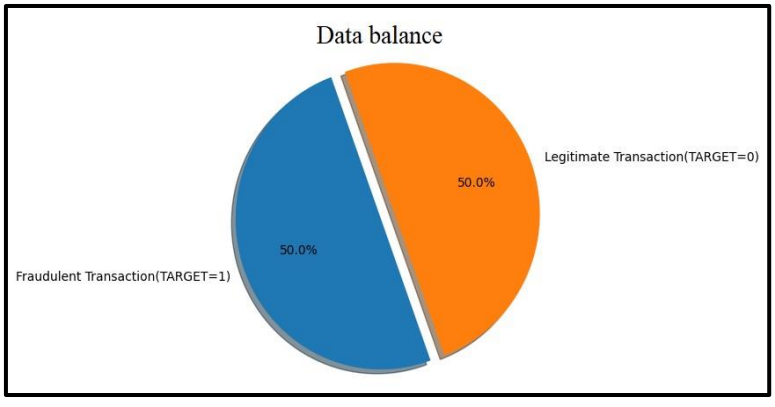

Figure (2). Data balancing.

The initial dataset, which skewered fraud cases, was rebalanced. The dataset had a revised distribution with a 50% fraud/50% non-fraud split. Class distribution equilibrium prevents model bias and ensures that machine learning models can distinguish fraudulent and non-fraudulent transactions.

Data balancing equalizes distribution through oversampling, as shown in the picture below. This crucial step strengthens our models and improves our fraud detection system.

5.2 The Classification Stage Results

The model's precision, recall, and F1 score were carefully evaluated. The weighted average of these metrics, depending on their support values (i.e., the number of true occurrences for each classification), was also investigated. Results from LR, KNN, NB, GB, DT, Adaboost, XGboost, LightGBM, and CatBoost using technique Oversampling, ROC, and AUC as in table (1) and figure (3-5).

Table (1) the experimental results of implementing classification algorithms with oversampling on the Fraudulent Transaction dataset.

Evolution / algorithm	Accuracy	Precision	Recall	F1-score
LR	95%	94%	96%	94%
KNN	98%	98%	99%	98%
NB	99%	99%	99%	99%
GB	98%	100%	96%	98%
DT	97%	99%	96%	97%
Adaboost	98%	100%	96%	98%
XGboost	98%	99%	96%	98%
LightGBM	98%	99%	96%	98%
CatBoost	98%	99%	96%	98%

Accuracy metrics matter in skewed datasets. These indicators work well together to choose the best model for unbalanced data. F1 scores are used to find a balance between precision and recall. The table shows that DT has the highest F1 score, followed by KNN. DT's highest grade is F1 (99%). The application detects fraud using the DT model.

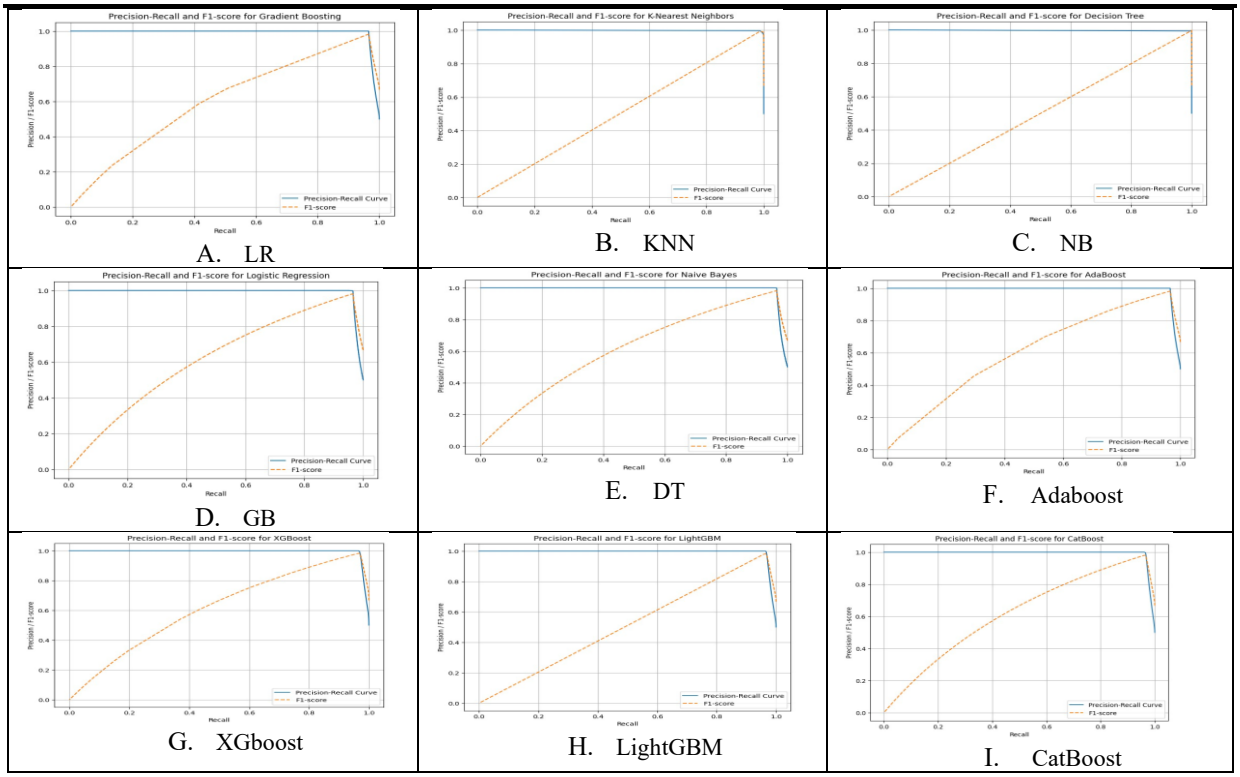

Figure (3) Show the Precision, Recall, and F1-score

The logistic regression (LR) model demonstrates a noteworthy level of accuracy. Its precision and recall metrics indicate a well-balanced performance in accurately detecting positive cases while minimizing the occurrence of false positives and false negatives. The bKNN algorithm exhibits a high level of overall accuracy, which suggests its effectiveness in accurately categorizing cases. The observed excellent precision and recall metrics indicate a commendable equilibrium in effectively reducing both false positives and false negatives. The model denoted as NB exhibits superior performance in terms of accuracy, precision, and F1-score. The remarkable performance of the system indicates a high level of proficiency in properly detecting instances of fraudulent personal information, while effectively managing the trade-off between precision and recall.

The performance of GB is characterized by achieving optimal precision, signifying that its predictions of positive instances are highly accurate. Nevertheless, a comparatively lower recall indicates the possibility of overlooking favorable cases. DT demonstrates a strong ability to recall, hence demonstrating its efficacy in recording a significant proportion of good experiences. Nevertheless, a marginal decrease in precision entails a compromise that could potentially lead to an increased occurrence of false positives.The ensemble approaches, including Adaboost, XGBoost, LightGBM, and CatBoost, regularly demonstrate a high precision rate of 99-100%. This highlights their effectiveness in reducing the occurrence of false positives.

The recall rate was found to be 96%.The decreased recall observed in this context indicates a potential difficulty in accurately identifying and capturing all cases that are positive. However, it is important to take into account the high precision associated with these events.

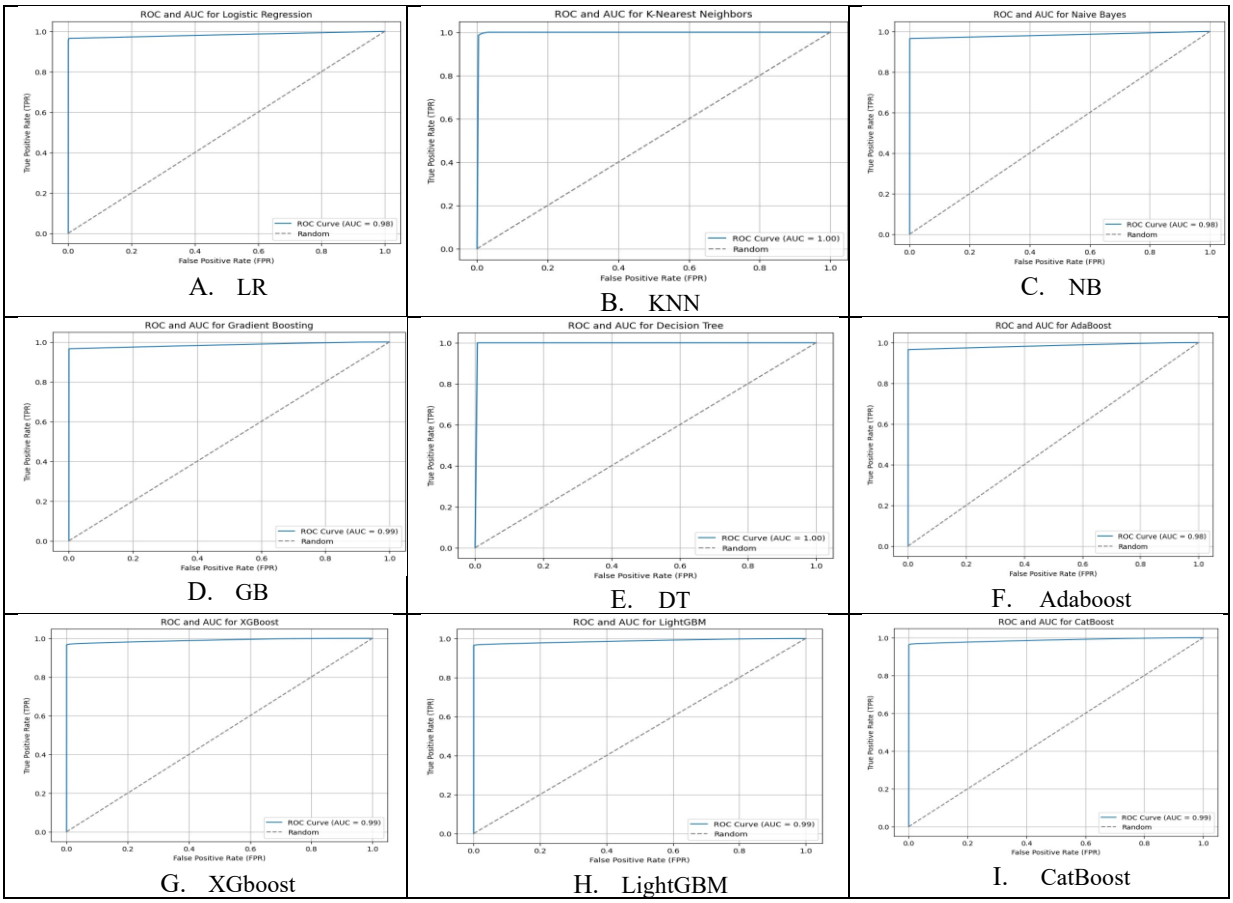

Figure (4) Show the ROC and AUC

To evaluate model performance, the ROC curve was plotted for all models. The Area and ROC curves The findings of the AUC (Under the ROC Curve) analysis show how effective the assessed machine learning algorithms are at identifying fraudulent personal information used in financial transactions. With an AUC of 0.94 and a well-shaped ROC curve, Logistic Regression (LR) exhibits a balanced trade-off between true positives and false positives. With an AUC of 0.98 and a stable ROC curve, KNN demonstrates its discriminatory capability. The ROC curves for both Gradient Boosting (GB) and Naive Bayes (NB) have AUC values of 0.99, indicating their sensitivity and accuracy. Decision Trees (DT) with effective discrimination have a stable ROC curve and an AUC of 0.99. The discriminatory performance of ensemble techniques such as Adaboost, XGBoost, LightGBM, and CatBoost is demonstrated by their well-shaped ROC curves and AUC values of 0.99. The models' capacity to differentiate between authentic and fraudulent transactions is demonstrated by the consistency of their high AUC values across several methods. Examine the needs of the application when selecting an algorithm; ensemble techniques show promise because of their reliable discriminatory capability.

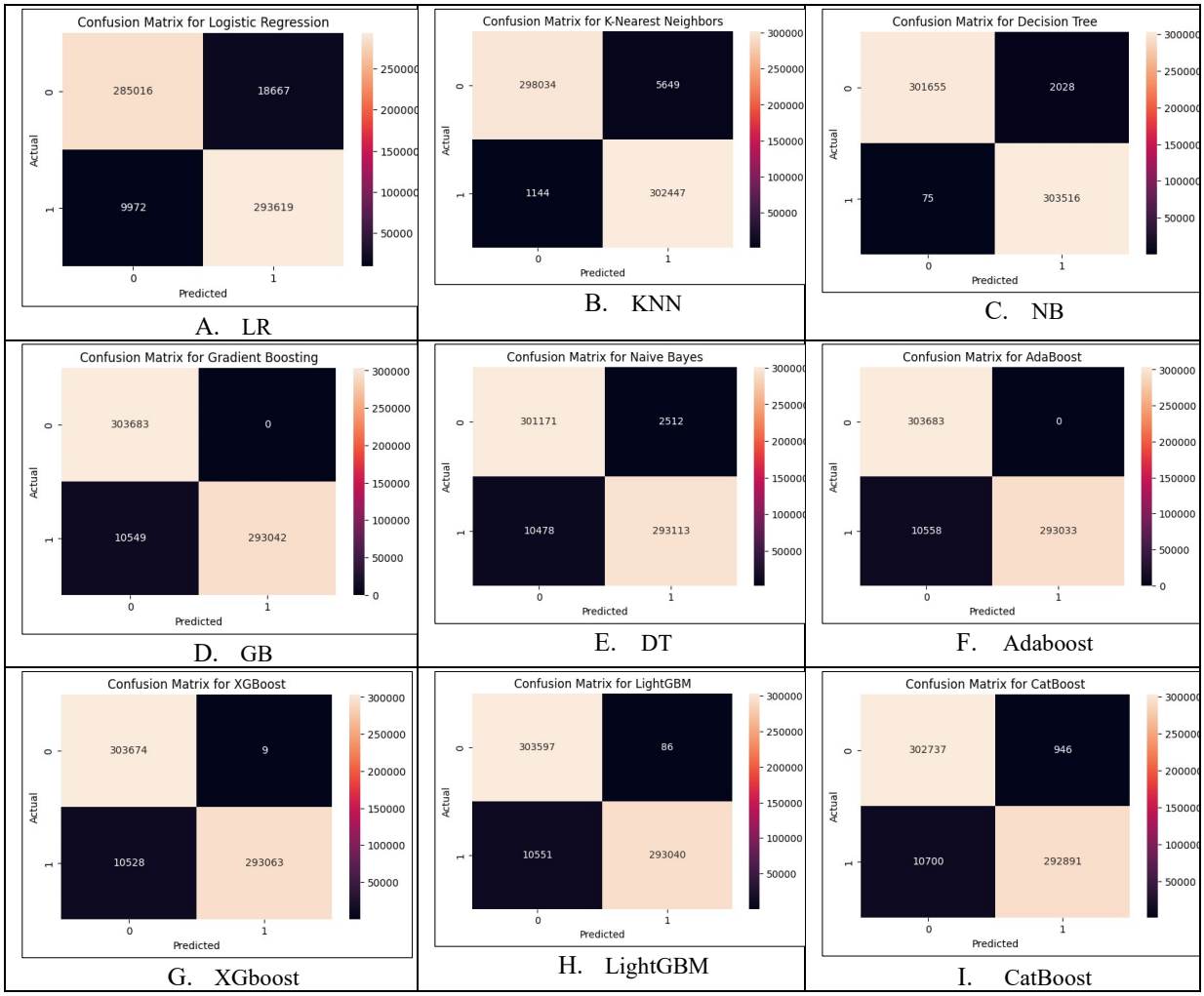

Figure (5) the confusion matrixes

Some algorithms have Precision-Recall trade-offs. Institutions may prioritize precision (minimizing false positives) or recall (recording as many positive examples as feasible) depending on the application. Ensemble Methods Dominance Ensemble approaches like Adaboost, XGBoost, LightGBM, and CatBoost perform well across several metrics.

Application Details Application needs should guide algorithm selection. To avoid inconveniencing legitimate users, fraud prevention may require precision to minimize false positives.

Finally, algorithm performance changes demonstrate the significance of using different measures for a complete review. Naive Bayes and ensemble approaches are good rivals, each with its own strengths. The final choice should match the financial application's requirements and limits.

5.3 Comparison of This Suggested System with Previous Studies

The table below presents a comparison between the newly suggested system and prior studies. Notably, the dataset exhibited significant class imbalance, with a mere 0.1345% of transactions categorized as fraudulent. We evaluated the effectiveness of 9 different machine learning algorithms in the task of fraud detection. Subsequently, we compared the performance of these

algorithms, with the decision tree emerging as the top performer, achieving an impressive accuracy rate of 99.86%.

Table (2) Shows a comparison of this suggested system with previous studies

No.	Ref.	Year	Method used	Performance
1.	[7]	2020	Bidirectional Long Short-Term Memory and Bidirectional Gated Recurrent Unit & machine learning algorithms	Their model outperformed machine learning classifiers with an accuracy score of 91.37%.
2.	[8]	2020	logistic regression, naive Bayes, decision trees and Artificial Neural Network (ANN)	An accuracy of 94.84% with logistic regression, 91.62% with naive Bayes, and 92.88% using decision trees. Our Artificial Neural Network (ANN) surpassed all other algorithms with 98.69% accuracy.
3.	[9]	2021	NB, LR, KNN, RF, and Sequential Convolutional Neural Network	accuracy scores of 96.1%, 94.8%, 95.89%, 97.58%, and 92.3% corresponding to NB, LR, KNN, RF, and Sequential Convolutional Neural Networks, respectively
4.	[10]	2014	Machine learning algorithms.	Their conclusion emphasized the ineffectiveness of imbalanced classification when dealing with highly skewed data, noting that existing methods were expensive and prone to false alarms.
5.	[11]	2018	DT, LR, and RF algorithms	The accuracy rates were LR 90.0%, RF 95.5%, and DT 94.3%
6.	[12]	2020	RF, SVM, and LR	found that RF performed exceptionally well, achieving the highest accuracy
7.	[13]	2017	SVM, RF, DT, and LR	Accuracy rates were SVM 97.5%, RF 98.6%, DT 95.5%, and LR 97.7%.
8.	[14]	2021	RF, SVM, and KNN	The accuracy rates for RF, SVM, and KNN are 94.84%, and 89.46%, and, notably, Random Forest exhibits swift detection of new fraud cases.
9.	[15]	2019	Support vector machine (SVM)	It achieved a fraud detection score of 91% using SVM.
10.	[16]	2019	deep neural network approach	The results demonstrated superior performance compared to classical models like SVM and LR.
11	[17]	2020	RF, SGD, and J48 classifiers	The accuracy percentages for Task1 and Task2 datasets were within the range of 97.78% to 98.1%, The RF classifier showed the highest kappa statistic value.
12	[18]	2020	DT and Rough Set methods	The accuracy rate of 84.25%

13	[19]	2016	Logistic Regression, SVM, and Xgboost. Lightgbm	Lightgbm has an accuracy rate of 98%, Logistic Regression (92.60%), SVM (95.20%), and Xgboost (97.10%).
14	[20]	2020	REDBSCAN and SVM algorithms	The SVDD achieved an AUC of 97.75%, while SVM achieved 94.60%.
15	[21]	2018	RF and KNN techniques	They used KNN to train a Dynamic Random Forest model to predict transaction outcomes cheaply.
16	[22]	2019	Naïve Bayes, Multilayer Perceptron, Adaptive Boosting, Gradient Boosting, and Random Forest	Every classifier is executed in a two-stage process, and each of them exhibits strong performance.
17	[23]	2020	decision trees XGBoost and LightGBM	The proposed algorithms outperform traditional classifiers with a classification quality of 0.977-0.982, reducing model synthesis time and identifying up to 85.7% of fraudulent transactions.
11.	The Proposed System	2023	LR, KNN, NB, GB, DT, Adaboost, XGboost, LightGBM, CatBoost	The accuracy was LR 95.28%, KNN 98.88%, DT 99.65%, GB98.26%, NB 97.86%, Adaboost 98.26%, XGboost 98.26%, LightGBM 98.24%, CatBoost 98.08%

Conclusions

The comparative study on machine learning techniques' efficiency in preventing fraudulent personal information in financial transactions highlights their importance in financial ecosystem security. Our Kaggle dataset analysis revealed compelling insights into the performance of Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Trees (DT), Gradient Boosting (GB), Naive Bayes (NB), Adaboost, XGBoost, LightGBM, and CatBoost. Decision Trees (DT) had the greatest accuracy rate of 99.65%, followed by K-Nearest Neighbours (KNN) at 98.88%, demonstrating their effectiveness in detecting bogus personal information fraud. Logistic Regression (LR), Naive Bayes (NB), Adaboost, XGBoost, LightGBM, and CatBoost also had high accuracy rates of 95.28% to 98.26%, strengthening detection and preventive methods. The study underlines the importance of using advanced machine learning methods to build robust systems that can detect complex financial transaction patterns and abnormalities. These accuracy rates provide valuable insights, but to ensure seamless integration into real-world financial systems, these algorithms should also consider computational efficiency, scalability, and interpretability. Our comprehensive evaluation shows the crucial role of machine learning algorithms in combating the misuse of fake personal information, providing valuable guidance for financial institutions seeking to strengthen their defenses against fraud and ensure financial transaction integrity and security. Increased research on optimizing and fine-tuning these algorithms will improve their performance and resilience in the ever-changing financial security context.

References

[1] M. M. M. Megdad, B. S. Abu-Nasser, and S. S. Abu-Naser, "Fraudulent Financial Transactions Detection Using Machine Learning," International Journal of Academic Information Systems Research, vol. 6, no. 3. pp. 30–39, 2022, [Online]. Available: www.ijeais.org/ijaisr.

[2] K. Vuppula, "An-advanced-machine-learning-algorithm-for-fraud-financial-transaction-detection.pdf." JIDPTS, 2021.

[3] T. Amarasinghe, A. Aponso, and N. Krishnarajah, "amarasinghe2018.pdf." Association for Computing Machinery, 2018, doi: https://doi.org/10.1145/3231884.3231894.

[4] M. A. Khan et al., "Application of Machine Learning Algorithms for Sustainable Business Management Based on Macro-Economic Data: Supervised Learning Techniques Approach." MDPI, 2022, doi: https:// doi.org/10.3390/su14169964.

[5] T. Pham and S. Lee, "Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods." 2016, [Online]. Available: http://arxiv.org/abs/1611.03941.

[6] M. R. Baker, Z. N. Mahmood, and E. H. Shaker, "Ensemble Learning with Supervised Machine Learning Models to Predict Credit Card Fraud Transactions." 2022, doi: https://doi.org/10.18280/ria.360401.

[7] M. Khedmati, M. Erfani, and M. GhasemiGol, "Applying support vector data description for fraud detection," arXiv Prepr. arXiv2006.00618, 2020.

[8] V. S, "Credit Card Fraud Detection using Machine Learning Algorithms," Int. J. Eng. Res., vol. V9, Aug. 2020, doi: 10.17577/IJERTV9IS070649.

[9] A. Mehbodniya et al., "Financial Fraud Detection in Healthcare Using Machine Learning and Deep Learning Techniques," Secur. Commun. Networks, vol. 2021, pp. 1–8, Sep. 2021, doi: 10.1155/2021/9293877.

[10] K. R. Seeja and M. Zareapoor, "Fraudminer: A novel credit card fraud detection model based on frequent itemset mining," Sci. World J., vol. 2014, 2014.

[11] S. Lakshmi and S. D. Kavilla, "Machine learning for credit card fraud detection system," Int. J. Appl. Eng. Res., vol. 13, no. 24, pp. 16819–16824, 2018.

[12] Y. Lucas et al., "Towards automated feature engineering for credit card fraud detection using multi-perspective HMMs," Futur. Gener. Comput. Syst., vol. 102, pp. 393–402, 2020.

[13] N. Carneiro, G. Figueira, and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," Decis. Support Syst., vol. 95, pp. 91–101, 2017.

[14] N. Kousika, G. Vishali, S. Sunandhana, and M. A. Vijay, "Machine Learning based Fraud Analysis and Detection System," J. Phys. Conf. Ser., vol. 1916, no. 1, p. 012115, May 2021, doi: 10.1088/1742-6596/1916/1/012115.

[15] V. N. Dornadula and S. Geetha, "Credit card fraud detection using machine learning algorithms," Procedia Comput. Sci., vol. 165, pp. 631–641, 2019.

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

[17] K. Abdulsattar and M. Hammad, Fraudulent Transaction Detection in FinTech using Machine Learning Algorithms. 2020.
[18] X. Yu, X. Li, Y. Dong, and R. Zheng, "A deep neural network algorithm for detecting credit card fraud," in 2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 2020, pp. 181–183.
[19] R. Jain, B. Gour, and S. Dubey, "A hybrid approach for credit card fraud detection using rough set and decision tree technique," Int. J. Comput. Appl., vol. 139, no. 10, pp. 1–6, 2016.
[20] D. Ge, J. Gu, S. Chang, and J. Cai, "Credit card fraud detection using lightgbm model," in 2020 international conference on E-commerce and internet technology (ECIT), 2020, pp. 232–236.
[21] S. Akila and U. S. Reddy, "Cost-sensitive Risk Induced Bayesian Inference Bagging (RIBIB) for credit card fraud detection," J. Comput. Sci., vol. 27, pp. 247–254, 2018.
[22] Y. Jain, N. Tiwari, S. Dubey, and S. Jain, "A comparative analysis of various credit card fraud detection techniques," Int. J. Recent Technol. Eng., vol. 7, no. 5, pp. 402–407, 2019.
[23] A. Mints and P. Sidelov, "Automatic machine learning algorithms for fraud detection in digital payment systems," Eastern-European J. Enterp. Technol., vol. 5, p. 14, Oct. 2020, doi: 10.15587/1729-4061.2020.212830.
[24] L. Li, Y. Zhang, and Y. Zhao, "k-Nearest Neighbors for automated classification of celestial objects," Sci. China Ser. G Physics, Mech. Astron., vol. 51, no. 7, pp. 916–922, Jul. 2008, doi: 10.1007/s11433-008-0088-4.
[25] O. F.Y, A. J.E.T, A. O, H. J. O, O. O, and A. J, "Supervised Machine Learning Algorithms: Classification and Comparison," Int. J. Comput. Trends Technol., vol. 48, no. 3, pp. 128–138, Jun. 2017, doi: 10.14445/22312803/IJCTT-V48P126.
[26] F. E. Botchey, Z. Qin, and K. Hughes-Lartey, "Mobile money fraud prediction—a cross-case analysis on the efficiency of support vector machines, gradient boosted decision trees, and naïve bayes algorithms," Information, vol. 11, no. 8, p. 383, 2020.
[27] S. Ruder, "An overview of gradient descent optimization algorithms," arXiv Prepr. arXiv1609.04747, 2016.
[28] P. Xu, "Review on Studies of Machine Learning Algorithms," J. Phys. Conf. Ser., vol. 1187, no. 5, p. 052103, Apr. 2019, doi: 10.1088/1742-6596/1187/5/052103.
[29] E. Yaman and A. Subasi, "Comparison of bagging and boosting ensemble machine learning methods for automated EMG signal classification," Biomed Res. Int., vol. 2019, 2019.
[30] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-Augu, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.
[31] M. Onoja, A. Jegede, J. Mazadu, G. I. O. Aimufua, A. Oyedele, and K. Olibodum, Exploring the Effectiveness and Efficiency of LightGBM Algorithm for Windows Malware Detection. 2022.
[32] A. Sharma, "Understanding GOSS and EFB: The core pillars of LightGBM." Towards Data Science. https://towardsdatascience. com/what-makes-lightgbm …, 2018.
[33] A. Joshi, P. Saggar, R. Jain, M. Sharma, D. Gupta, and A. Khanna, "CatBoost- An Ensemble Machine Learning Model for Prediction and Classification of Student Academic

Performance," Adv. Data Sci. Adapt. Anal., vol. 13, Oct. 2021, doi: 10.1142/S2424922X21410023.
[34] Z. S. Kadhim, H. S. Abdullah, and K. I. Ghathwan, "Artificial Neural Network Hyperparameters Optimization : A Survey," vol. 18, no. 15, pp. 59–87, 2022.
[35] A. S. Issa, Y. H. Ali, and T. A. Rashid, "Comparative Analysis of Swarm Algorithms to Classification of covid19 on X-Rays," 2022 Int. Conf. Data Sci. Intell. Comput. ICDSIC 2022, no. Icdsic, pp. 164–169, 2022, doi: 10.1109/ICDSIC56987.2022.10075733.
[36] A. S. Issa, Y. H. Ali, and T. A. Rashid, "An Efficient Hybrid Classification Approach for COVID-19 Based on Harris Hawks Optimization and Salp Swarm Optimization," vol. 18, no. 13, pp. 113–130.