Enhancing Real-Time Credit Card Fraud Detection Using Stacking Ensemble Methods

1st Nicol Buratti
University of Camerino
Department of Computer Science
Camerino, Italy
nicol.buratti@studenti.unicam.it

2nd Damiano Pasquini
University of Camerino
Department of Computer Science
Camerino, Italy
damiano.pasquini@studenti.unicam.it

3rd Mathukiya Vaibhav Jagdish
University of Camerino
Department of Computer Science
Camerino, Italy
vaibhavjag.mathukiya@studenti.unicam.it

***Abstract*—Credit card fraud detection has become a pressing concern in the digital age, with the rise of online transactions and the increasing sophistication of fraudsters. Recent studies have focused on leveraging advanced machine-learning techniques to improve detection accuracy. Various approaches have been developed to face this problem, such as deep learning techniques as demonstrated in [1], and transformer model [2]. The paper presents some machine learning applications to detect credit card fraud with high accuracy and efficiency. Specifically, in this paper, we propose a framework that consists of various machine learning algorithms, such as logistic regression, decision trees, and random forests, among others popularly used, integrated using the stacking ensemble method, to analyze transaction data for fraudulent activities, outperforming the state-of-the-art results. This paper examines the comparative advantages of our models in relation to existing implementations outlined in prior literature. Additionally, we explore the integration of our final model into real-time transaction processing systems. The conclusion offers suggestions for future research directions, including the detection of specific fraud types and the identification of emerging fraud trends. This study investigates the effectiveness of machine learning in credit card fraud detection, and how it surpasses traditional human monitoring with significant enhancements. Through the utilization of sophisticated algorithms and comprehensive datasets, financial institutions can enhance customer protection and mitigate financial losses attributable to fraudulent transactions.**

***Index Terms*—Fraud Detection, Anomaly Detection, Machine Learning**

I. Introduction

Credit card fraud represents a significant challenge in the modern financial landscape, imposing substantial economic losses and compromising consumer trust. The latest statistics released from Federal Trade Commission data show that consumers reported losing more than 10$ billion to fraud in 2023, making it the first time that fraud losses have reached that benchmark. This marks a 14% increase over reported losses in 2022 [10]. As digital transactions increase, the sophistication and frequency of fraudulent activities have surged, necessitating the development of advanced detection mechanisms. An important reason why this research is being developed is the need to reduce the time and research spent on fraud management. Another main goal is to reduce the number of false positives, which are transactions that are mistakenly identified as fraudulent. This is important because it can lead to a loss of revenue for the company using a specific payment circuit. This research aims to develop a model that can accurately predict fraudulent transactions. The model will be trained on a dataset of credit card transactions, named *Credit Card Fraud Detection Dataset 2023* made available by the authors on Kaggle [5] as detailed in Section III. In previous works, machine learning algorithms, particularly supervised learning methods, are extensively utilized to analyze transaction patterns and identify potential fraudulent activities. Commonly used algorithms include Decision Trees, Random Forests, Logistic Regression, Support Vector Machines, and Neural Networks. These methods excel in identifying complex, non-linear relationships in large datasets. According to Carcillo et al. [3], ensemble methods, such as Random Forests and Gradient Boosting, have demonstrated high accuracy in fraud detection by combining the predictions of multiple models to improve robustness and reduce overfitting. Similarly, supervised learning techniques like Logistic Regression and Support Vector Machines effectively distinguish between fraudulent and legitimate transactions due to their ability to handle large feature spaces and provide probabilistic outputs according to Bhattacharyya et al. [4]. In our study, we initially performed exploratory data analysis to understand the dataset's characteristics by doing preprocessing, feature selection, and feature engineering, and we verified that the dataset classes were balanced. We then inspire our approach from the ensemble model Khalid et. al. [6] proposed in their work, integrating more machine learning algorithms described in detail in Section III. We utilized hyperparameter tuning to optimize the model's performance and evaluated its accuracy, precision, recall, and F1 score. Then we evaluate the model's performance using a pipeline for each of them. A complication we faced was determining where overfitting was occurring in most of the models, and we made use of the cross-validation technique to mitigate this issue. Our contribution to the field of fraud detection is the development of a robust, high-performing model that can accurately predict fraudulent transactions, outperforming existing models in terms of accuracy and efficiency. The model's high accuracy and low false positive rate will enable financial institutions to detect fraudulent activities more effectively, reduce economic losses,

and enhance consumer trust. The rest of the paper is organized as follows: Section II provides an overview of related works in the field of credit card fraud detection. Section III describes the dataset, methodology, and evaluation metrics used in this study. Section IV presents the results of our experiments, and discusses the findings. Finally, Section V concludes the paper and outlines future research directions.

II. Related work

The problem of credit card fraud detection has been extensively studied in the literature, with researchers exploring various machine-learning techniques to address this challenge. In this section, we present some of the most relevant works in the field. Classical approaches such as Gradient Boosting (GB), Support Vector Machines (SVM), Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF) have proven useful.

One of the early and most important works in this domain was by Khalid et. al. [6] where they proposed an ensemble of machine learning models to detect credit card fraud. They used a combination of Support Vector Machines (SVM), K-Nearest Neighbour (KNN), Random Forest (RF), Gradient Boosting (GB), and AdaBoost to detect fraud. They used the Kaggle dataset for their experiments and achieved an accuracy of 99.9%. They also compared their results with other machine learning models and found that the ensemble model outperformed them. In their work they identified limitations in the existing technologies, including issues like data imbalance, concept drift, false positives/negatives, limited generalizability, and challenges in real-time processing - to address these issues, they developed as said before an ensemble model that combines the strengths of multiple machine learning models.

Another relevant work was by Seera, Manjeevan, et al. [7] where they present an intelligent payment card fraud detection system that utilizes machine learning techniques to effectively identify fraudulent transactions. The researchers developed a fraud detection model that combines supervised and unsupervised learning algorithms. They used a dataset of credit card transactions to train and evaluate the model, which included features such as transaction amount, merchant category, and user behavior patterns. The system demonstrated high accuracy in identifying fraudulent transactions, with a detection rate of over 95% and a low false positive rate. The model was able to adapt to changing fraud patterns and maintain its performance over time. The proposed system can be effectively deployed by financial institutions to enhance the security of payment card transactions and protect customers from financial losses due to fraudulent activities. The intelligent, adaptive nature of the system makes it a valuable tool in the fight against evolving payment card fraud schemes.

Building upon this, Abhimanyu, et al. [8] present a deep-learning approach for detecting fraudulent credit card transactions. The researchers developed a neural network model that can accurately identify fraudulent transactions by learning patterns in large datasets of historical credit card activity. Their results show that the deep learning model outperforms traditional machine learning techniques in terms of fraud detection accuracy and speed. The authors propose that this deep learning framework could be implemented by credit card companies and financial institutions to enhance their fraud prevention capabilities in real-time. They also suggest exploring the integration of additional data sources, such as customer location and purchase history, to further improve the model's predictive power. Overall, the paper demonstrates the potential of deep learning to revolutionize credit card fraud detection and provide more robust security for financial transactions.

In a similar vein, El Kafhali et al. [9] propose an optimized deep learning approach to detect fraudulent transactions by leveraging a combination of techniques to improve the accuracy and efficiency of fraud detection. The approach involves using a Long Short Term Memory network (LSTM), a convolutional neural network (CNN) to extract relevant features from transaction data, which are then fed into a recurrent neural network (RNN) to capture temporal patterns. The output from the RNN is then passed through a fully connected neural network to generate a final prediction. To optimize the approach, the authors employ several techniques, including data augmentation, transfer learning, and ensemble methods. The proposed approach is evaluated on a large dataset of real-world transactions and is shown to significantly outperform existing methods in terms of both accuracy and computational efficiency, making it a promising solution for detecting fraudulent transactions in real time.

III. Materials and Methods

A. Dataset

In this study, we utilized the "*Credit Card Fraud Detection Dataset 2023*" [5], which is publicly available on Kaggle [1]. This dataset encompasses more than 550,000 credit card transactions conducted by European cardholders during the year 2023.

The dataset comprises 31 features, with 29 of them anonymized to safeguard the identities of the cardholders. The "Amount" feature represents the transaction amount, while the final feature, "Class", serves as the transaction label indicating whether a transaction is fraudulent (1) or non-fraudulent (0). Importantly, the dataset is balanced, with 50% of transactions labeled as fraudulent and the remaining 50% as non-fraudulent.

B. Preprocessing

Data preprocessing is a crucial step in machine learning, as it ensures that the data is clean, consistent, and ready for model training. In our study, we initially performed Exploratory Data Analysis (EDA) to gain insights into the dataset's structure and characteristics. During this process, we examined the distribution of features, identified missing values, and assessed the balance of the dataset. We found that the dataset is

[1]Dataset repository: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023

perfectly balanced, with an equal number of fraudulent and non-fraudulent transactions. In order to perform this analysis we used the *pandas*, *matplotlib*, and *sci-kit learn* libraries.
Following EDA, we proceeded with data preprocessing, which involved scaling the features with several techniques, including Standard Scaler, Min-Max Scaler, Recursive Feature Elimination (RFE), and Principal Component Analysis (PCA). Important to notice is that using the PCA we reduced the number of features from 29 to 12.

C. Experiment

Our approach drew inspiration from the stacking ensemble model proposed in [6], where we evaluated several powerful models within this framework. These models included Logistic Regression (LR), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes (NB), Decision Trees (DT), and Random Forest (RF).
To determine the most effective model for our dataset, we constructed a pipeline for each of the proposed algorithms. Each pipeline scales the training data before feeding it into the model for training. Subsequently, the test data is used to generate a classification report, which evaluates the model's effectiveness based on various performance metrics, as we can see in Table I.

	precision	recall	f1-score
Decision tree	0.9982	0.9982	0.9982
Logistic Regression	0.9628	0.9621	0.9621
Gaussian Naive Bayes	0.9241	0.9182	0.9178
Bernoulli Naive Bayes	0.2495	0.5000	0.3329
Random Forest	0.9999	0.9999	0.9999
K-Nearest Neighbors	0.9976	0.9976	0.9976

TABLE I
PROPOSED MODELS CLASSIFICATION REPORT

As illustrated in Table I, the Decision Tree, Logistic Regression, Random Forest, and K-Nearest Neighbors algorithms demonstrated superior performance on our dataset. Given our goal of achieving real-time prediction with high accuracy, we decided to discard one model between K-Nearest Neighbors and Random Forest due to their slower prediction times, in order to simplify the model while maintaining sufficient precision. We selected Random Forest for its lower variance, which contributes to more consistent performance.
Consequently, we will focus on these three models for further analysis and evaluation.

D. Hyper-parameters tuning

We developed our stacking model by combining predictions from the Decision Tree and Random Forest using Logistic Regression, which is well-suited for binary classification problems. To enhance the model's performance, we fine-tuned the hyper-parameters of both the Decision Tree and Random Forest models. These optimized models were then used to fine-tune the hyperparameters of the Logistic Regression component within the stacking model, thereby improving overall predictive accuracy. The flow diagram of this phase is illustrated in Figure 1.

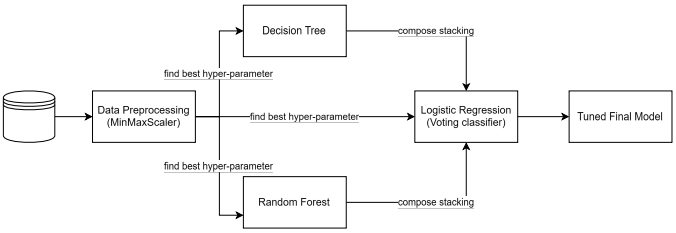

Fig. 1. Flow Diagram of the Machine Learning model tuning

To achieve the optimal hyperparameters, we utilized the *GridSearchCV* object provided by the scikit-learn library. This object conducts an exhaustive search over specified parameter values for an estimator, employing a cross-validation splitting strategy. We configured a 5-fold cross-validation with the following parameters, highlighting the optimal values in green:

parameters	1	2	3	4	5
max_depth	None	10	20	30	40
min_samples_split	2	10	20	30	
max_features	None	sqrt	log2		

TABLE II
PARAMETERS TESTED ON DECISION TREE

parameters	1	2	3
n_estimators	100	200	
max_depth	None	10	20
min_samples_split	2	10	
max_features	None	sqrt	log2
bootstrap	True	False	

TABLE III
PARAMETERS TESTED ON RANDOM FOREST

parameters	1	2	3
penalty	l1	l2	
C	0.1	1.0	10.0
solver	liblinear	saga	

TABLE IV
PARAMETERS TESTED ON LOGISTIC REGRESSION IN THE STACKING

The use of *GridSearchCV* enabled us to systematically explore a wide range of parameter combinations, ensuring that the final model was fine-tuned to achieve the best possible performance. This approach allowed us to identify the most effective hyper-parameter settings for each model component, thereby optimizing the overall performance of the stacking model. During this process, each candidate model was evaluated based on its performance across multiple folds of the dataset, ensuring that the final model was robust and not overly dependent on any single subset of the data. This rigorous approach helped mitigate the risk of overfitting and improved the generalization capabilities of the model.

E. Final model training

After identifying the optimal hyper-parameters for the final model, as illustrated in Figure 2, we proceeded with splitting our dataset into two distinct subsets: a training sample

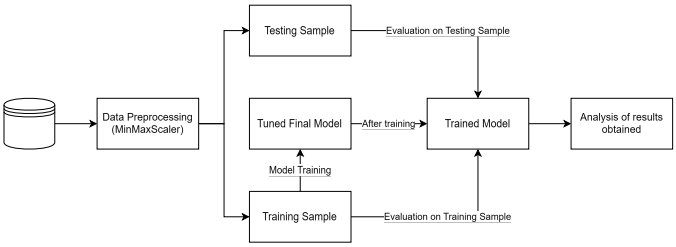

Fig. 2. Training process of the final model

comprising 80% of the data and a test sample comprising the remaining 20%. The primary purpose of this division was to ensure that our model could be effectively trained and subsequently evaluated on separate data to validate its performance. The training sample was used to build and train our model, leveraging the fine-tuned hyperparameters obtained through the GridSearchCV process.

Once the model was trained, we employed both the training and test samples to assess its performance comprehensively. The training sample allowed us to monitor how well the model learned from the data it was trained on, ensuring that it was not overfitting or underfitting. On the other hand, the test sample provided an unbiased evaluation of the model's generalization ability, reflecting its performance on unseen data. We generated detailed reports and performed an in-depth analysis of the results from both the training and test samples. These reports included various performance metrics such as accuracy, precision, recall, F1-score, and others, which provided insights into the model's effectiveness and reliability. By thoroughly analyzing these results, we aimed to confirm the robustness of our model and its suitability for real-time prediction with high accuracy.

IV. Results and Discussion

In this section, we present the performance evaluation of our machine learning models—Decision Tree, Random Forest, and Stacking Ensemble—for detecting credit card fraud. The effectiveness of each model is assessed using key metrics such as precision, recall, and F1-score. We employed cross-validation techniques to ensure the robustness and reliability of our results, providing a comprehensive view of each model's performance. All the written code is available in GitHub [2]

A. Decision Tree

The decision tree model demonstrated a notable accuracy of 99.8% in predicting credit card fraud. The confusion matrix in Figure 3 provides a detailed view of the model's performance, showing the true positives, true negatives, false positives, and false negatives. This matrix helps in understanding the types of errors the model makes and its overall prediction capability. Metrics:

- Precision: 99.8%

[2]Project repository: https://github.com/damiano00/credit_card_fraud_detection

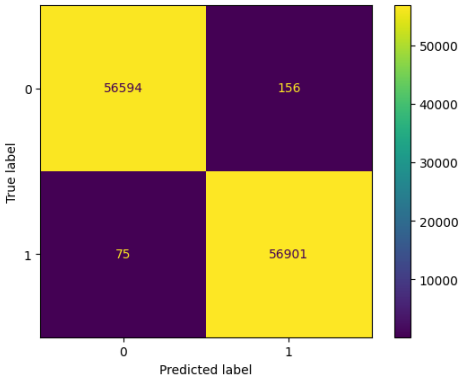

Fig. 3. Confusion matrix of the Decision Tree

- Recall: 99.9%
- F1-score: 99.8%

To ensure the model's robustness, we applied 10-fold cross-validation. The average accuracy from this process was 99.7%, with a standard deviation of 0.09.

B. Random Forest

The Random Forest model exhibited an impressive accuracy of 99.8% in predicting credit card fraud. The confusion matrix in Figure 4 provides insights into the model's performance metrics, revealing a balanced distribution of prediction errors.4. Metrics:

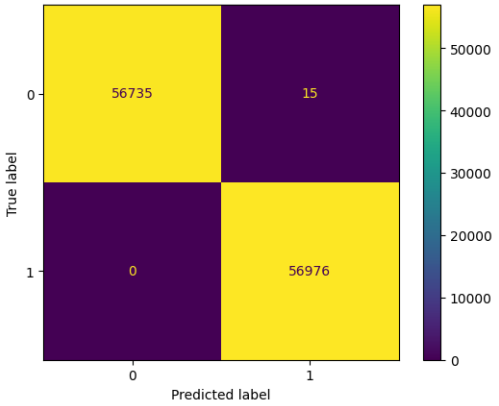

Fig. 4. Confusion matrix of the Random Forest

- Precision: 99.9%
- Recall: 99.9%
- F1-score: 99.9%

The cross-validation process yielded an average accuracy of 99.9%, with a minimal standard deviation of 0.0002. This demonstrates the model's high reliability and robustness in detecting fraudulent transactions.

C. Stacking

The Stacking ensemble model, which combines the predictions of the Decision Tree and Random Forest models using Logistic

Regression, achieved an accuracy of 99.8% in predicting credit card fraud. The confusion matrix in Figure 5 provides a detailed overview of the model's classification performance.

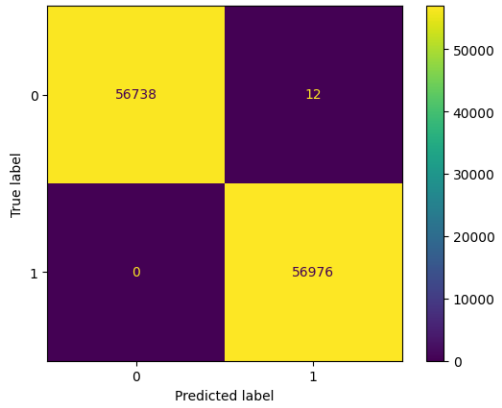

Fig. 5. Confusion matrix of the Stacking model

Metrics:

- Precision: 100%
- Recall: 100%
- F1-score: 100%

Cross-validation results for the stacking model showed an average accuracy of 99.9%, with a standard deviation of 0.0002. This indicates that the stacking approach effectively harnesses the strengths of its component models to deliver superior performance.

D. Prediction Latency

Prediction latency is a critical factor in evaluating the effectiveness of a real-time credit card fraud detection system. Low latency ensures that the model can process and analyze transactions quickly, enabling immediate detection of fraudulent activities. This subsection focuses on evaluating the prediction latency of the proposed machine learning models. In order to gauge this latency, experiments were conducted using a subset of our dataset to minimize waiting times.

Two distinct experiment methodologies were employed: individual transaction prediction timing and batch transaction prediction timing. As demonstrated, the per-transaction time in the batch method is reduced due to optimizations implemented.

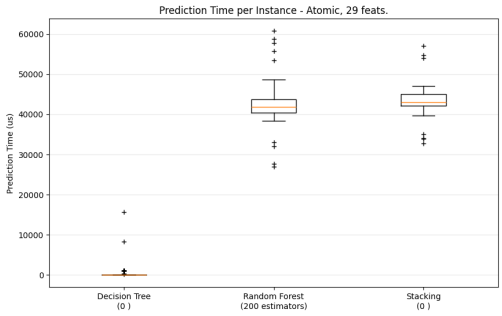

Fig. 6. Atomic latency of the three models

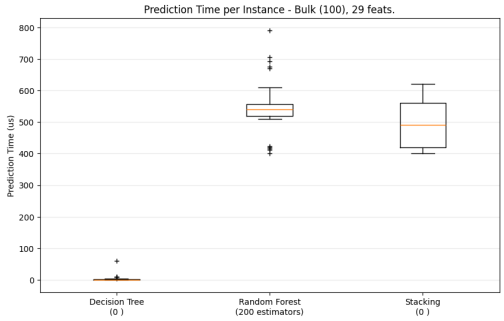

Fig. 7. Bulk latency of the three models

As depicted in Figure 6 and Figure 7, the decision tree model demonstrates significantly lower latency when contrasted with both random forest and stacking methods. Despite its initial average prediction time of 45ms, which can be optimized to 0.5ms, this performance aligns well with our project's objectives.

To further reduce prediction latency, we implemented a strategy to decrease the number of features used during predictions. We conducted experiments by training the stacking model on our sample dataset, each time randomly selecting a subset of features.

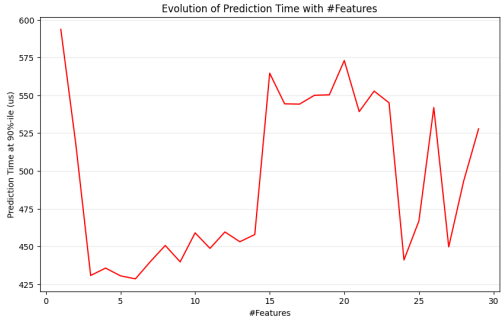

Fig. 8. Prediction time per number of features

As shown in Figure 8, using methods such as PCA (Principal Component Analysis) or RFE (Recursive Feature Elimination) can provide more refined approaches to improving prediction latency compared to randomly selecting a subset of features. These techniques systematically identify and prioritize features based on their relevance and contribution to the model's predictive accuracy. By implementing PCA, we can reduce the dimensionality of the feature space while retaining essential information. Similarly, RFE iteratively selects features based on their impact on model performance, ensuring that only the most informative features are utilized. This optimization can significantly enhance both prediction speed and overall model performance.

E. Discussion

The experimental results highlight the effectiveness of machine learning models in detecting credit card fraud with high precision and recall. The Decision Tree and Random Forest models

individually performed exceptionally well, but the stacking ensemble model provided a slight edge in terms of overall accuracy and reliability. Our findings are consistent with the literature, which underscores the efficacy of ensemble methods in enhancing prediction accuracy by leveraging the strengths of multiple models. The high precision and recall rates indicate that our models can accurately distinguish between fraudulent and non-fraudulent transactions, significantly reducing the risk of false positives and negatives. The minimal standard deviations observed in the cross-validation results suggest that our models are robust and generalize well across different subsets of the data. This robustness is crucial for real-world applications, where the model must perform reliably on unseen data. Overall, the stacking model's superior performance makes it a promising candidate for integration into real-time fraud detection systems. By combining the predictions of multiple well-tuned models, the stacking approach mitigates the limitations of individual models and enhances the overall detection capability.

V. Conclusion

The study presented a comprehensive approach to credit card fraud detection using advanced machine learning techniques. By employing a variety of algorithms such as Logistic Regression, Decision Trees, and Random Forests, and integrating them through a stacking ensemble method, we achieved a high level of accuracy and efficiency in detecting fraudulent transactions. The findings underscore the superiority of ensemble methods, particularly the stacking model, which demonstrated an exceptional ability to predict fraudulent activities with minimal false positives and negatives. The rigorous preprocessing, feature engineering, and hyperparameter tuning contributed significantly to the robustness and reliability of the models. The proposed framework not only outperformed existing models but also showed promise for real-time application, enhancing financial institutions' capability to protect customers and mitigate economic losses.

Future research should explore the detection of specific fraud types and adapt the models to emerging fraud trends, ensuring continuous improvement in fraud detection systems. Moreover, they should focus on enhancing the detection of specific types of fraud by incorporating domain-specific features and refining algorithms to adapt to the evolving nature of fraudulent activities. This includes developing models that can identify and adapt to new and emerging fraud patterns, ensuring the system remains robust against sophisticated attacks. Exploring the application of advanced deep learning techniques, including but not limited to Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), could yield significant improvements in detection accuracy. These models are adept at capturing temporal patterns and intricate relationships within data, making them suitable for real-time fraud detection.

References

[1] Du, HaiChao, et al. "A novel method for detecting credit card fraud problems." Plos one 19.3 (2024): e0294537.

[2] Yu, Chang, et al. "Credit Card Fraud Detection Using Advanced Transformer Model." arXiv e-prints (2024): arXiv-2406.

[3] Carcillo, Fabrizio, et al. "Combining unsupervised and supervised learning in credit card fraud detection." Information sciences 557 (2021): 317-331.

[4] Bhattacharyya, Siddhartha, et al. "Data mining for credit card fraud: A comparative study." Decision support systems 50.3 (2011): 602-613.

[5] Credit Card Fraud Detection Dataset 2023. Kaggle. https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023

[6] Khalid, Abdul Rehman, Nsikak Owoh, Omair Uthmani, Moses Ashawa, Jude Osamor, and John Adejoh. 2024. "Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach" Big Data and Cognitive Computing 8, no. 1: 6. https://doi.org/10.3390/bdcc8010006

[7] Seera, Manjeevan, et al. "An intelligent payment card fraud detection system." Annals of Operations Research 334.1 (2024): 445-467.

[8] Roy, Abhimanyu, et al. "Deep learning detecting fraud in credit card transactions." 2018 systems and information engineering design symposium (SIEDS). IEEE, 2018.

[9] El Kafhali, Said, Mohammed Tayebi, and Hamza Sulimani. "An Optimized Deep Learning Approach for Detecting Fraudulent Transactions." Information 15.4 (2024): 227.

[10] ftc.gov. (2024, February 9). As nationwide fraud losses top $10 billion in 2023, FTC steps up efforts to protect the public. Federal Trade Commission. https://www.ftc.gov/news-events/news/press-releases/2024/02/nationwide-fraud-losses-top-10-billion-2023-ftc-steps-efforts-protect-public