Utkal University Journal of Computing & Communications,
Vol. 2, Issue. 1 (2024) *e-ISSN:2584-203X*

Tracking Fraudulent Transactions in Credit Cards using Logistic Regression

Soumen Nayak[1], Sipra Sahoo[1], Chinmayee Mund[1], Priyanshu Satpathy[1], Ankita Swain[1], Anwesha Mahapatra[1]

[1]Department of Computer Science and Engineering, Siksha 'O' Anusandhan (Deemeed to be) University, Bhubaneswar, Odisha, India

soumennayak@soa.ac.in, siprasahoo@soa.ac.in, 1941012038.g.priyanshusatpathy@gmail.com, 1941012651.g.chinmayeemund@gmail.com, 1941012251.g.ankitaswain@gmail.com, 1941012079.g.anweshamahapatra@gmail.com

Abstract: E-commerce has reshaped the global trade front, now becoming an all-important tool for organizations, businesses, and governments to increase effectiveness. A primary reason driving the growth of e-commerce is the ease of card transactions through the Internet. However, with the spread of digital payments, cybercrime cases and cybersecurity issues also increased. Fraud financing deliberately denies the victim's rights or gains unlawful monetary advantages. Thus, detecting fraud is one of the primary concerns for banks and other financial institutions. ML is emerging as an innovative solution in detecting credit card fraud, surpassing conventional techniques by identifying complex patterns within big data. By analyzing user behavior, payment methods, and transaction patterns, it can predict and prevent abnormal activities. This study assesses the capability of ML algorithms to identify fraudulent credit card transactions through the application of algorithms like Logistic Regression (LR), Random Forest (RF), and K-Nearest Neighbors (KNN). Through credit card transactions in Kaggle data, accuracy, precision, and F1-score assessments are evaluated on these models. According to the outcome, the model that depicted maximum accuracy is LR while attaining 94% for fraud detection accuracy. This shows the potential of ML techniques to enhance fraud detection systems and provide more security and efficiency in digital payment systems. The findings also highlight the importance of embracing advanced analytics in combating financial fraud and securing the ever-changing e-commerce landscape.

Keywords: Machine learning, Fraudulent transactions, Credit cards, Logistic regression, Accuracy

1. Introduction

Credit card fraud has become a pressing issue in the digital age, significantly impacting financial institutions and individuals. With the surge in online transactions and the sophistication of fraudulent schemes, traditional fraud detection systems have struggled to keep pace. Rule-based systems, where the patterns are predefined, face high false positive rates and fail to detect new fraud tactics and evolving ones that cause significant financial and reputational losses [2]. Therefore, the need for innovative and accurate fraud detection mechanisms has become a

✉ soumennayak@soa.ac.in

priority.

This paper applies ML techniques to overcome credit card fraud detection challenges. Logistic Regression (LR), K-Nearest Neighbors (KNN), and Random Forest (RF) were selected for their proven effectiveness in classification tasks and their potential to enhance fraud detection accuracy [4]. This can enable the fast processing of large datasets containing transactional data to discover complex patterns and obtain predictions in real-time, making them a robust alternative to current systems [3]. In addition, several automated machine learning frameworks have emerged as crucial solutions, making the development process and deployment of such an algorithm faster while maintaining high precision [9].

One central goal of this work involves testing and comparing the algorithms based on precision, recall, F1-score, or accuracy. Another key research objective is to handle an issue in fraud detection that often arises in datasets: class imbalance. Some other methods, such as undersampling and synthetic data generation, are used to resolve the problem and make a model more reliable [7]. Further, other cutting-edge methods, like feature selection and hyperparameter optimization, are also applied to improve the performance of algorithms and scalability [5].

The practical results of this research are massive. Financial institutions can adopt this knowledge to improve fraud detection frameworks, reduce operational costs, and enhance customer trust [8]. Besides, it is a foundation for future research by promoting hybrid approaches and sophisticated feature engineering techniques. It is towards filling the gap between classical and modern detection systems, protecting stakeholders from ever-emerging fraud threats.

Many studies have emphasized the significance of machine learning in fraud detection. For instance, [6] pointed out that data-driven models are highly relevant to identifying fraud patterns, and [10] demonstrated the capability of ensemble techniques to boost detection accuracy. Aleskerov et al. [1] opened the field of neural networks in fraud detection, but their contribution paved the way for later developments in the field. These results clearly show the impact of ML algorithms on preventing financial fraud.

This paper is structured as follows: Section 2 reviews relevant literature. Section 3 discusses the background, problem statement, and research questions. The proposed methodology is given in Section 4. Section 5 describes the model evaluation and discussion. Finally, Section 6 concludes with insights and suggestions for future work.

2. Literature Survay

Fraud is an illegal or unlawful deception intended to produce financial or personal gain. It is a deliberate violation of a law, regulation, or policy with the intent of acquiring unauthorized financial benefits. The literature discussed below focuses on different machine-learning techniques and models used for fraud detection, especially in the credit card fraud domain.

Randhawa et al. [11] reviewed and compared popular algorithms, including Deep Learning (DL), Support Vector Machine (SVM), and Naive Bayes (NB). They compared hybrid and standard models incorporating AdaBoost and majority voting techniques using public credit card data. The majority vote had the highest MCC score of 0.823. When testing hybrid models, data samples were subjected to 10% to 30% noise.

Raj et al. [12] applied the machine learning techniques of XGBoost, Decision Tree (DT), K-Nearest Neighbor (KNN), Random Forest (RF), Logistic Regression (LR), and SVM on a dataset of online credit card transactions. They evaluated the algorithms using the confusion matrix, F1, and accuracy scores. The results showed that all six models successfully detected fraud since they have high accuracy.

In its research, Awoyemi et al. [13] evaluated the performance of the use of Logistic Regression (LR), K-Nearest Neighbor (KNN), and Naive Bayes (NB) on highly imbalanced data related to credit card fraud. The dataset comprised 284,807 European cardholders' transactions, using a hybrid under/over-sampling strategy. Performance

metrics such as precision, sensitivity, Matthews' correlation, accuracy, balanced classification rate, and specificity metrics are used to get the result. KNN showed its best accuracy at 97.69%, while Naive Bayes scored 97.92%. The lowest accuracy score stood at 54.86% for LR.

Xuan et al. [14] employed a pair of Random Forest (RF) classifiers, namely CART-based RF and Random-tree RF, to identify fraud and expected behavior. Based on a dataset from a Chinese e-commerce business, their work compared the performance of the RF classifiers using different base models.

Thennakoon et al. [15] used SVM, NB, and LR for fraud detection. SVM was reported to achieve the highest accuracy rate of 91%, outperforming NB and LR. Their future work focuses on fraud detection based on location.

Ahmed et al. [16] proposed an alert algorithm known as Intimation Rule-Based (IRB). They developed an ontology-based fraud detection tool to identify and prevent financial fraud, generating alerts with varying severity levels and eliminating dead notifications for improved efficiency and reliability.

Wang et al. [17] optimized the Back Propagation (BP) neural network using the Whale Optimization Algorithm (WOA). This solved problems like system instability and slow convergence. The WOA provided the optimal initial values, further refined by the BP network algorithm for better performance.

Malini and Pushpa [18] utilized outlier detection methods and the KNN algorithm to track fraudulent transactions. Experiment results showed that KNN was more efficient and accurate than other anomaly detection techniques. Their approach also aimed to reduce the misclassification of genuine transactions as fraudulent.

Bonkoungou et al. [19] compared the performances of classifiers LR, RF, and GB for fraud detection. Due to frequent changes in pattern, the best algorithm was still chosen, RF.

Varmedja et al. [20] used SMOTE to oversample an imbalanced dataset used for credit card fraud. They tested Multilayer Perceptron, RF, LR, and NB methods and concluded that all could determine fraud accurately.

Reddy et al. [21] used XGBoost, LDR, and Gradient-based classifiers. XGBoost had the highest accuracy in fraud detection, though results were strictly dependent on data values. This caused repetition issues.

Yee et al. [22] applied five Bayesian classifiers (K2, Logistics, NB, J48, and TAN) for fraud detection. Cleansed data showed improved results considerably for all classifiers, while performance degraded when tested in other datasets.

Trivedi et al. [23] used machine learning models: ANN, NB, RF, LR, tree classifiers, and SVM to detect fraud. Among them, RF achieved maximum accuracy. However, the authors still suggested potential improvements in the efficiency of classifiers.

Shen et al. [7] analyzed the performances of LR, neural networks, and decision trees for detecting fraud. LR and neural networks outperformed decision trees due to their excellent problem-solving capabilities.

Adepoju et al. [25] tested NB, SVM, LR, and KNN. LR gave the best accuracy but required parameter tuning for the best performance.

Alfaiz and Fati [26] used several models: CatBoost, LR, NB, DT, KNN, RF, LightGBM, GBM, and XGBoost. AllKNN with CatBoost (AllKNN-CatBoost) provided the best results, yet they depended on datasets.

Ileberi et al. [27] experimented with decision trees, SVM, extreme gradient boosting, RF, extra trees, and AdaBoost. XGB-AdaBoost outperformed other algorithms, although the class imbalance problem led to diminished classification quality.

Boutaher et al. [28] studied supervising algorithms, such as SVM, RF, and LR. They concluded that high utilization of supervised algorithms could be recommended while highlighting a need for extensive performance metrics to help determine the best models.

Yousefi et al. [29] used LR, ANN, DT, SVM, and NB for fraud detection. LR was the most efficient model, but performance was reduced with the size of the dataset.

Pradhan et al. [30] experimented with ANN, RF, and other algorithms for fraud detection. ANN was the most accurate, and multiple calculations as modules were suggested to improve the final accuracy.

Mienye and Jere [31] reviews DL-based approaches like CNN, RNN, LSTM, and GRU, compares their performance, addresses challenges in training fraud models, and demonstrates their robustness using real-world datasets.

Baria et al. [32] proposed hybrid method combining deep learning's ability to capture complex patterns with linear regression's interpretability for effective fraud detection. A deep learning model, such as CNN or RNN, extracts feature from transaction data, which are then classified using linear regression. This approach enhances performance while offering insights into fraud, supporting financial institutions in combating credit card fraud.

The dynamic shopping patterns of credit card holders and class imbalance challenge ML classifiers' performance. Mienye and Sun [33] addresses these issues with a deep learning ensemble combining LSTM and GRU networks in a stacking framework with MLP as the meta-learner, alongside the SMOTE-ENN method for class balancing. Experimental results demonstrate superior sensitivity (1.000) and specificity (0.997), outperforming traditional ML classifiers.

Despite these advancements, existing techniques often need help with problems, including data imbalance, overfitting, and scalability issues. Many models require extensive tuning and are sensitive to dataset variations. This study addresses these gaps by leveraging state-of-the-art algorithms like Random Forest, Logistic Regression, and K-Nearest Neighbors on real-world datasets, focusing on feature selection, hyperparameter optimization, and imbalanced learning techniques to improve detection accuracy and generalizability.

3. Background

This section discusses existing training Models on Machine Learning followed by the definition of the problem statements of the proposed model.

3.1. Random Forest

Random Forest is an ensemble learning technique that combines multiple decision trees to enhance prediction accuracy. It creates a "forest" of decision trees, each contributing to the final prediction. The output of the Random Forest model is determined by majority voting for classification tasks or averaging for regression tasks.

The steps of the Random Forest algorithm are as follows:

i. Randomly sample data points (with replacement) to create subsets of the original dataset (Bootstrap sampling).
ii. Build a decision tree for each subset by selecting a random subset of features at each split.
iii. Repeat this process to create multiple trees.
iv. Predict outcomes for a new instance by having all trees vote or average their outputs.
v. The final prediction is based on the class with the highest votes (classification).

The Random Forest algorithm can be mathematically represented as follows:

For a dataset D = $\{(x_1, y_1), (x_2, y_2), \dots\dots, (x_n, y_n)\}$ where x_i represents features and y_i represents labels.

i. Randomly select N subsets $D_1, D_2, \dots\dots, D_b$ with replacement.
ii. For each subset D_b, build a decision tree $H_B(X)$ using k-random features at each node split.
iii. Aggregate predictions from all trees: $\hat{Y} = mode\{h_1(x), h_2(x), \dots\dots, h_b(x)\}$ for classification

3.2. Logistic Regression

Logistic Regression is a statistical method used for binary classification problems. It estimates the probability of a target variable belonging to a particular class based on input features. The model uses a logistic (sigmoid) function to map predicted values to probabilities within the range [0, 1].

The logistic regression model can be expressed as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-z}}$$

Where

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Here:

- $P(Y = 1|X)$: Probability of the target variable being 1 given the feature set X.
- $\beta_0, \beta_1, \ldots, \beta_n$: Model coefficients.
- $x_1, x_2, \ldots x_n$: Input features.

The steps of the Logistic Regression algorithm are:

1. Split the dataset into training and testing sets.
2. Fit the logistic regression model on the training set by finding the coefficients (β) using a likelihood-based optimization.
3. Use the fitted model to compute probabilities for the test set using the sigmoid function.
4. Classify an instance as class 1 if $P(Y = 1|X) \geq 0.5$, otherwise class 0.

3.3. K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) is a non-parametric and instance-based learning algorithm for classification and regression tasks. The core idea of KNN is to classify a data point based on the majority class of its K-nearest neighbors in the feature space.

The steps of the KNN algorithm are:

1. Store all the training data points.
2. For a new data point x′, calculate the distance $d(x', x_i)$ between x′ and each training point x_i.
    - The most common distance metric is Euclidean distance:

$$d(x', x_i) = \sqrt{\sum_{j=1}^{n} \left(x'_j - x_{ij}\right)^2}$$

3. Identify the K-nearest neighbors to x′.
4. Assign the majority class label (for classification).

Advantages of KNN:

- Simple to implement and intuitive.
- Effective for small datasets with fewer features.

However, KNN requires careful selection of K, as too small a value may lead to overfitting, while a large K may over-smooth the classification boundary.

3.4. Problem Statement

Detecting credit card fraud remains a critical challenge due to the low accuracy and biased solutions often associated with traditional approaches. These limitations lead to substantial financial losses for credit card companies and cardholders. A significant obstacle is the highly imbalanced nature of the dataset, where fraudulent transactions constitute a small fraction of the total, making accurate detection difficult. Traditional machine-learning models need help with such imbalanced datasets, often favoring the majority class (legitimate transactions) and failing to detect fraudulent ones effectively. There is a pressing need for improved techniques to handle imbalanced datasets while achieving higher accuracy and fairness in detecting fraudulent transactions.

This study aims to evaluate the performance of Logistic Regression against advanced machine learning models such as Random Forest and K-Nearest Neighbor (K-NN) in detecting credit card fraud. Additionally, it investigates the influence of dataset characteristics, including imbalance and size, on the models' performance. This work tries to find the solutions to the research questions given below:

- How does Logistic Regression perform in detecting fraudulent transactions compared to other machine learning algorithms like Random Forest and K-NN?
- What specific challenges do imbalanced datasets pose in the context of fraud detection, and how can Logistic Regression be adapted to address these challenges?
- How does the size of the training dataset impact the performance of Logistic Regression, Random Forest, and K-NN in detecting fraudulent transactions?

These research questions aim to explore the limitations of existing methods, evaluate alternative algorithms, and propose strategies to improve fraud detection outcomes.

4. Proposed Methodology

The identification of fraudulent credit card transactions by combining the discussed training models is proposed in this section with the input description.

4.1. Dataset Description

In this paper, the dataset used is fetched from Kaggle (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud) in CSV format. It's European credit card transaction data from two days. There are 284,315 legitimate transactions and 492 fraudulent ones; the percentage of fraudulent transactions comes to 0.172%, making the cases much more challenging to detect the frauds precisely.

The dataset features only numerical input variables derived through a Principal Component Analysis (PCA) transformation as shown in the Figure 1. Due to confidentiality constraints, the original attributes and additional sensitive details have been withheld. The primary features, labeled as $V_1, V_2 \ldots\ldots, V_{28}$ are the components generated by PCA. Two attributes, **Time** and **Amount**, have not undergone PCA transformation:

- **Time**: Represents the elapsed time in seconds between each transaction and the first transaction in the dataset.
- **Amount**: Reflects the transaction value in monetary terms.

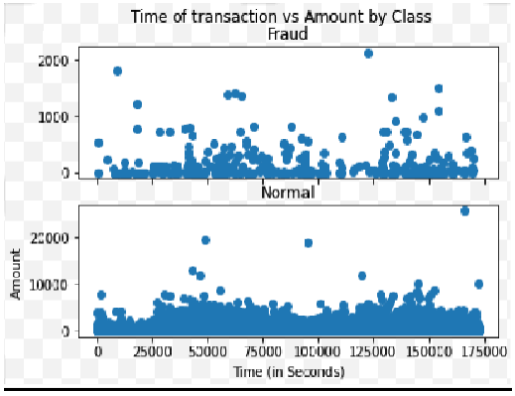


Fig 1. Scatter Plot of the Dataset

The target variable, **Class**, is binary and indicates the nature of the transaction:

- 1: Fraudulent transaction
- 0: Legitimate transaction

This dataset provides an excellent foundation for developing and evaluating machine learning models to address the challenges of fraud detection in highly imbalanced data scenarios.

4.2. Flowchart of the Proposed Model

.The flowchart in Figure 2 outlines the schematic layout of the system for detecting credit card fraud. It represents the sequential data processing flow, algorithm application, and decision-making stages in identifying fraudulent transactions. The primary components of the architecture include:

Data Source:

- The system begins with a dataset containing transaction records (e.g., obtained from Kaggle).
- It includes both fraudulent and legitimate transactions, with anonymized features and sensitive details withheld for confidentiality.

Data Preprocessing Module:

- **Normalization**: Ensures uniform scaling of features such as "Time" and "Amount."
- **Class Balancing**: Tackles the issue of class imbalance through oversampling (e.g., SMOTE) or undersampling techniques.
- **Feature Engineering**: Utilizes PCA-transformed features (V1 to V28) alongside unaltered features like "Time" and "Amount."

Model Training and Testing:

- **Training Data**: A portion of the preprocessed dataset is used to train the models.
- **Machine Learning Algorithms**: Models such as **Random Forest**, **Logistic Regression**, and **K-Nearest Neighbor (K-NN)** are implemented.
- **Evaluation Metrics**: Metrics like accuracy, precision, recall, and F1-score are used to assess model performance.

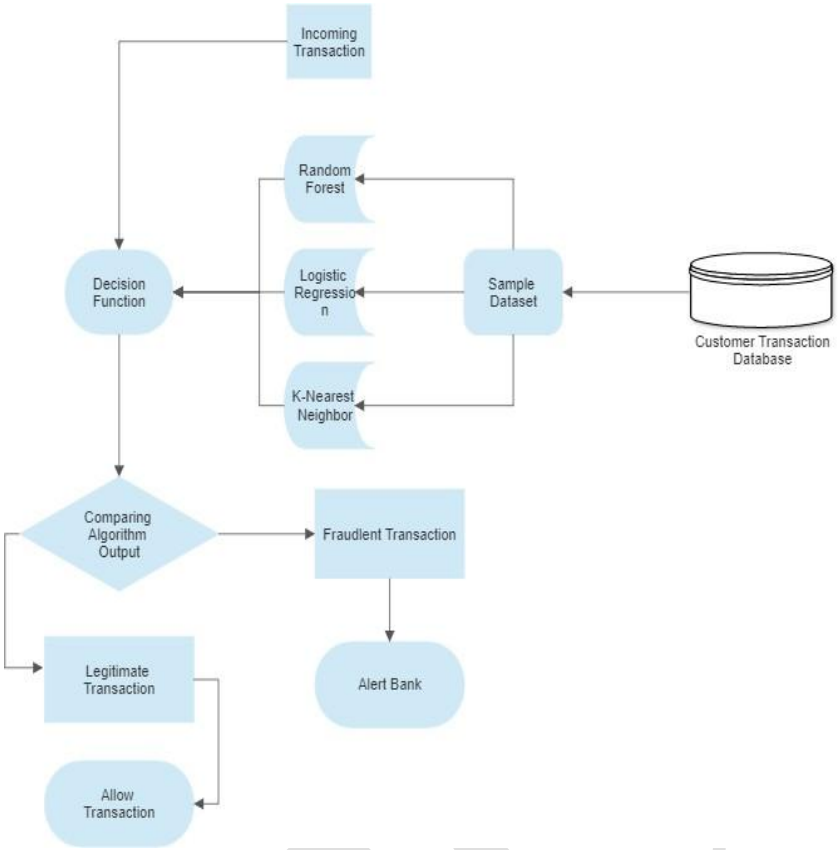


Fig 2. Schematic Layout of the Proposed System

Fraud Detection Engine:

- The trained models process incoming transaction data.
- Each model provides predictions on whether a transaction is fraudulent (Class=1) or legitimate (Class=0).

Decision-Making Layer:

- Aggregates predictions from multiple models using a voting mechanism or another ensemble strategy to finalize the outcome.

Output Module:

- **Flagged Transactions**: Fraudulent transactions are flagged for further scrutiny.
- **Legitimate Transactions: Allowed to proceed without interruption, ensuring smooth operations for cardholders.**

This architecture ensures a streamlined and effective process for detecting credit card fraud, balancing accuracy and computational efficiency while safeguarding financial transactions. The fundamental architecture diagram in Figure 3 provides a high-level overview of the system implemented to detect credit card fraud. It outlines the core components, data flow, and interaction between different modules, emphasizing the systematic approach to processing credit card transactions and identifying fraudulent activities.

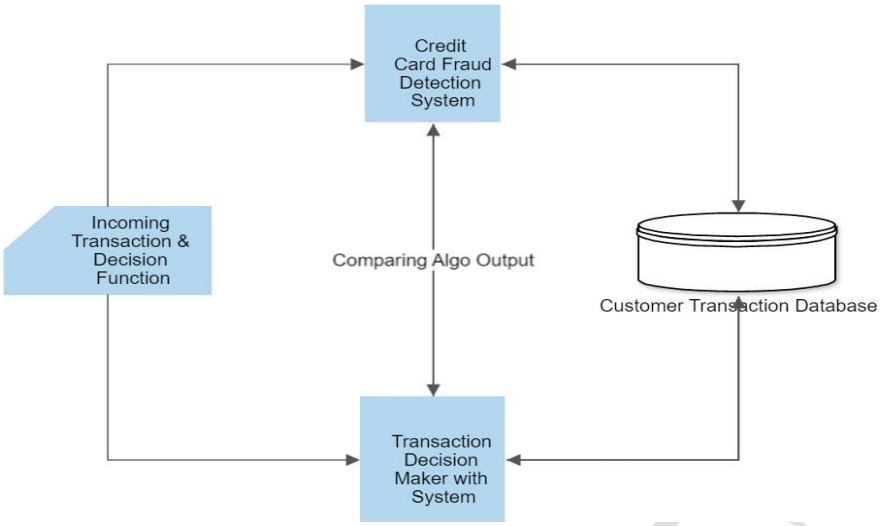


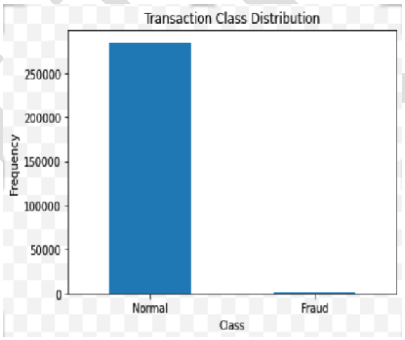
Fig 3. Fundamental Architecture of the Proposed System

5. Model Evaluation and Discussion

The proposed model preprocess the data, trained the model, and answered the research questions. This section put forth the confusion matrix followed by a discussion.

5.1. Class Imbalance and Balancing Techniques in Data Preprocessing

Machine learning algorithms often need help to achieve high performance when classification categories are not evenly distributed. The dataset used in this study is inherently imbalanced, with most transactions being legitimate and a minority being fraudulent. Balancing techniques are essential to ensure that the model is trained effectively.
In this study, under-sampling was applied to the legitimate transactions to balance the dataset effectively. Figures 4 and 5 illustrate the dataset distribution before and after balancing.



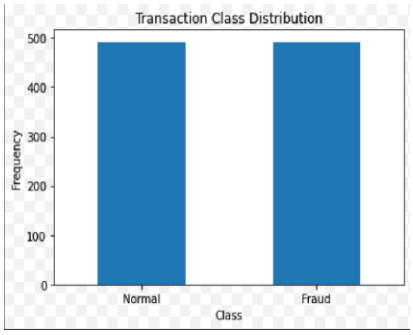


Fig 4. Imbalanced dataset **Fig 5. Balanced Dataset**

5.2. Dataset Splitting and Model Training

Once balanced, the dataset was split into two parts:

- **Training Set (80%)**: Used to train the machine learning models.
- **Testing and Validation Set (20%)**: Used to evaluate model performance.

Three classification models were built using **Logistic Regression**, **Random Forest**, and **K-Nearest Neighbor (KNN)**. These models were evaluated using comparative criteria to identify the most suitable algorithm for detecting

fraudulent transactions.

5.3. Performance Metrics

The models were assessed using several key metrics, calculated using a **Confusion Matrix**:

- **Accuracy**: The proportion of correctly identified transactions (both fraudulent and legitimate).
- **Precision**: The fraction of correctly identified fraudulent transactions among all transactions flagged as fraudulent.
- **Recall (Sensitivity)**: The fraction of actual fraudulent transactions correctly identified.
- **F1 Score**: The harmonic mean of precision and recall.

5.4. Confusion Matrix Results

By examining and comparing the confusion matrices of logistic regression, KNN, and random forest, one can make informed assessments of their individual capabilities in accurately classifying data. This visual representation aids in the evaluation and selection of the most suitable algorithm for a specific classification task.

Table 1 presents the confusion matrices results for the three models—**Random Forest (RF)**, **Logistic Regression (LR)**, and **K-Nearest Neighbor (KNN)**—highlighting the distribution of true positives, false positives, true negatives, and false negatives.

Table 1: Confusion matrix for several categorization models: Random Forest (RF), Logistic Regression (LR) and K-Nearest Neighbor (KNN)

		Predicted Legit	Predicted Fraud
Logistic Regression	Actual Legit	97	2
	Actual Fraud	15	83

		Predicted Legit	Predicted Fraud
K-NN	Actual Legit	68	23
	Actual Fraud	51	55

		Predicted Legit	Predicted Fraud
Random Forest	Actual Legit	95	4
	Actual Fraud	12	86

5.5. Classification Report and Model Comparison

Table 2 summarizes the F1 score, accuracy, precision, recall, and support for each model.
The results are categorized into:

- **Class 0 (Legitimate Transactions)**
- **Class 1 (Fraudulent Transactions)**

From a total of 492 fraudulent transactions, the following results were achieved:

- Logistic Regression outperformed with an accuracy rate of **94 %** in detecting fraud.
- The comparative results demonstrated the efficiency of Logistic Regression over KNN and Random Forest for this study.

Algorithm		Precision	Recall	F1-Score	Support	Accuracy
Logistic Regression	0	0.87	0.98	0.92	99	94 %
	1	0.98	0.85	0.91	98	
K-NN	0	0.57	0.75	0.65	91	62.4 %
	1	0.71	0.52	0.6	106	
Random Forest	0	0.89	0.97	0.93	99	92.89 %
	1	0.97	0.88	0.92	98	

Table 2: Comparing the F_1 Score, Accuracy, Precision, Recall, and Support for Logistic Regression, KNN, and Random Forest Models.

This study leverages machine learning algorithms to enhance the accuracy of fraud detection systems. This paper demonstrates that Logistic Regression achieves comparable or superior accuracy, proving its robustness in fraud detection tasks.

5.6. Discussion

This discussion addresses the research questions outlined in the Section 3.

Performance of logistic regression in detecting fraudulent transactions compared to K-NN and Random Forest

Logistic Regression (LR) performs better in detecting fraudulent transactions than K-NN and Random Forest. This is evidenced by its higher **F1 score**, **precision**, and **recall** values. Logistic Regression maintains a strong balance between detecting fraudulent (class 1) and non-fraudulent (class 0) transactions. It achieves a low **false negative rate** (failing to identify fraud) and a high **true positive rate** (correctly identifying fraud), showcasing its effectiveness in this domain. The clear separation provided by LR's probabilistic model helps it outperform the other methods in distinguishing legitimate and fraudulent activities.

Specific challenges posed by imbalanced datasets in detecting credit card fraud, and solution by Logistic Regression

Imbalanced datasets are a significant hurdle in detecting credit card fraud, as most transactions are legitimate (class 0), and fraudulent transactions (class 1) represent a tiny proportion. This imbalance skews algorithm training, causing models to favor the dominant class, leading to higher false negatives.
Logistic Regression can address these challenges through data preprocessing techniques, such as:

- **Under-Sampling**: Reducing the number of samples in the dominant class (legitimate transactions).
- **Over-Sampling**: Replicating or generating synthetic samples for the minority class (fraudulent transactions).
- **Hybrid Techniques**: Combining under-sampling and over-sampling for optimal class balance.

By employing these methods, Logistic Regression can train on a balanced dataset, improving its efficiency in identifying fraudulent transactions. Applying these preprocessing techniques ensures a fair representation of both classes and enhances the overall robustness of the model.

Impact of the training dataset size on the performance of Logistic Regression, K-NN, and Random Forest in detecting fraudulent transactions

The size of the training dataset plays a pivotal role in the performance of machine learning algorithms:

- **Logistic Regression**: This algorithm performs well with small to medium-sized datasets due to its simplicity and ability to handle high-dimensional data. Its efficiency remains consistent even with a limited training dataset, provided it is balanced.
- **Random Forest**: This ensemble method benefits from larger datasets, allowing its decision trees to learn diverse patterns. However, it remains robust even with smaller datasets due to its feature selection and averaging mechanisms.
- **K-Nearest Neighbor (K-NN)**: K-NN heavily relies on the size of the training dataset. A larger dataset provides more representative neighbors, improving classification accuracy. However, K-NN is computationally intensive with large datasets, which may lead to inefficiencies during prediction.

While larger datasets generally improve model performance by providing more representative patterns, a vast dataset may lead to overfitting, particularly for algorithms like Random Forest. Therefore, an optimal training dataset size is crucial to balance generalization and computational efficiency. The study confirms that Logistic Regression, combined with effective preprocessing techniques, emerges as the most reliable model for detecting fraudulent transactions, even in imbalanced datasets. Random Forest and K-NN also exhibit strengths in specific scenarios. Still, their reliance on data size and computational resources makes them less efficient than Logistic Regression in this particular application.

6. Conclusion

Credit card fraud is both a criminal and an important commercial issue, as it causes substantial financial and personal losses. Organizations are investing significantly in innovative methods to detect and prevent such fraudulent activities to overcome this challenge. This paper evaluated the performance of three machine learning classifiers: Logistic Regression, Random Forest, and K-Nearest Neighbors (KNN) in fraudulent transaction identification. The precision rates achieved were 94% for Logistic Regression, 62.4% for Random Forest, and 92.89% for KNN, with Logistic Regression performing the best in classifying transactions as fraudulent or legitimate. Though the study did not achieve 100% accuracy, it developed a model that performs reliably and is close to the goal. Future work would be to integrate more algorithms and explore ensemble learning techniques to optimize the precision of the model. Advanced resampling strategies such as SMOTE could help overcome the dataset's imbalance. Increasing the dataset with additional features and reducing false positives would increase the robustness and reliability of the model. Scaling the model for larger datasets and implementing it in real-time systems would allow dynamic fraud detection. Also, the proposed work should be compared with more existing methods in future for better justification of the work done. By addressing these aspects, further research can aim for higher accuracy and reliability to improve financial security and user trust.

References

1. Aleskerov, E., Freisleben, B., & Rao, B. (1997, March). Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)* (pp. 220-226). IEEE.

2. Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision support systems*, *50*(3), 602-613.
3. Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. *Information sciences*, *557*, 317-331.
4. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2017). Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, *29*(8), 3784-3797.
5. Krivko, M. (2010). A hybrid model for plastic card fraud detection systems. *Expert Systems with Applications*, *37*(8), 6070-6076.
6. Phua, C., Lee, V., Smith, K., & Gayler, R. (2010). A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.
7. Shen, A., Tong, R., & Deng, Y. (2007, June). Application of classification models on credit card fraud detection. In *2007 International conference on service systems and service management* (pp. 1-4). IEEE.
8. West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. *Computers & security*, *57*, 47-66.
9. Zöller, M. A., & Huber, M. F. (2021). Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, *70*, 409-472.
10. Sadineni, P. K. (2020, October). Detection of fraudulent transactions in credit card using machine learning algorithms. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 659-660). IEEE.
11. Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access*, *6*, 14277–14284.
12. Raj, S., Jain, M., & Chouksey, P. (2021). A Network Intrusion Detection System Based on Categorical Boosting Technique using NSL-KDD. *Indian Journal of Cryptography and Network Security*, *1*(2), 1-4.
13. Awoyemi, J., Adetunmbi, A. O., & Oluwadare, S. A. (2017). Credit card fraud detection using machine learning techniques: A comparative analysis. *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 1–9.
14. Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., & Jiang, C. (2018). Random forest for credit card fraud detection. *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, 1–6.
15. Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 488-493). IEEE.
16. Ahmed, M., Ansar, K., Muckley, C. B., Khan, A., Anjum, A., & Talha, M. (2021). A semantic rule-based digital fraud detection. *PeerJ Computer Science*, *7*, e649.
17. Wang, C., Wang, Y., Ye, Z., Yan, L., Cai, W., & Pan, S. (2018). Credit card fraud detection based on whale algorithm optimized BP neural network. *2018 International Conference on Computer Science and Education (ICCSE)*, 1–4.
18. Malini, N., & Pushpa, M. (2017). Analysis on credit card fraud identification techniques based on KNN and outlier detection. *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 255–258.
19. Bonkoungou, S., Roy, N. R., Ako, N. H. A., & Batra, U. (2023). Credit card fraud detection using ML: A survey. *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, 732–738.

20. Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019). Credit card fraud detection: Machine learning methods. *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 1–5.
21. Reddy, T. S., Nookaraju, G., Vikas, K., Mohanty, S. N., Anagandula, J., & Ahmed, M. S. (2022). An analysis of various algorithmic behaviors in detecting financial fraud. *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6.
22. Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as a data mining technique. *Journal of Telecommunication, Electronic and Computer Engineering*, *10*(1–4), 23–27.
23. Trivedi, N. K., Simaiya, S., Lilhore, U. K., & Sharma, S. K. (2020). An efficient credit card fraud detection model based on machine learning methods. *International Journal of Advanced Science and Technology*, *29*(5), 3414–3424.
24. Adepoju, O., Wosowei, J., & Jaiman, H. (2019). Comparative evaluation of credit card fraud detection using machine learning techniques. In *2019 Global Conference for Advancement in Technology (GCAT)* (pp. 1-6). IEEE.
25. Alfaiz, N. S., & Fati, S. M. (2022). Enhanced credit card fraud detection model using machine learning. *Electronics*, *11*(4), 662.
26. Ileberi, E., Sun, Y., & Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. *IEEE Access*, *9*, 165286-165294.
27. Boutaher, N., Elomri, A., Abghour, N., Moussaid, K., & Rida, M. (2020, November). A review of credit card fraud detection using machine learning techniques. In *2020 5th International Conference on cloud computing and artificial intelligence: technologies and applications (CloudTech)* (pp. 1-5). IEEE.
28. Yousefi, N., Alaghband, M., & Garibay, I. (2019). A comprehensive survey on machine learning techniques and user authentication approaches for credit card fraud detection. *arXiv preprint arXiv:1910.09546*.
29. Asif, M., Tauseef, S., & Khan, N. (2019). Comparative analysis of machine learning techniques for fraud detection in credit card transactions. *Proceedings of the 2019 International Conference on Data Science and Advanced Analytics (DSAA)*, 1–6.
30. Pradhan, S. K., Rao, N. K., Deepika, N. M., Harish, P., Kumar, M. P., & Kumar, P. S. (2021, December). Credit Card Fraud Detection Using Artificial Neural Networks and Random Forest Algorithms. In *2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1471-1476). IEEE.
31. Mienye, I. D., & Jere, N. (2024). Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions. *IEEE Access*.
32. Baria, J. B., Baria, V. D., Bhimla, S. Y., Prajapati, R., Rathva, M., & Patel, S. (2024). Deep Learning based Improved Strategy for Credit Card Fraud Detection using Linear Regression. *Journal of Electrical Systems*, *20*(10s), 1295-1301.
33. Mienye, I. D., & Sun, Y. (2023). A deep learning ensemble with data resampling for credit card fraud detection. *IEEE Access*, *11*, 30628-30638.