IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ML - BASED CREDIT CARD FRAUD DETECTION

[1] Prof. Dinesh Deore, [2]Kaushal Pethkar, [3]Abdullah Qureshi, [4]Ajinkya Bhosale, [5]Ali Khan

Department of Artificial Intelligence and Data Science, Rizvi College of Engineering, Mumbai, India.

Abstract: Credit card fraud presents a persistent and growing challenge in the modern financial landscape, with increasingly sophisticated fraudulent schemes leading to substantial financial losses for institutions and consumers. As digital transactions become more prevalent, the need for advanced, intelligent fraud detection mechanisms becomes paramount. Traditional rule-based systems often fail to adapt to new fraud strategies, necessitating more flexible and adaptive approaches. Machine learning (ML), with its ability to learn from historical data and generalize to unseen patterns, offers a promising solution for detecting fraudulent credit card transactions. In this study, we explore the effectiveness of various ML algorithms—namely Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and Neural Networks—in identifying fraudulent transactions within a highly imbalanced dataset. We employ extensive data pre-processing techniques, such as normalization and dimensionality reduction via PCA, and use SMOTE to address class imbalance. Our evaluation relies on multiple performance metrics including precision, recall, F1-score, and AUC-ROC to provide a comprehensive assessment of each model. Results indicate that ensemble methods, particularly Random Forest, outperform other techniques in both accuracy and reliability. This paper underscores the critical role of ML in enhancing fraud detection systems and suggests future research directions including real-time implementation and the integration of deep learning models for improved adaptability and efficiency.

Keywords: Credit Card Fraud, Machine Learning, Classification, Imbalanced Data, Random Forest

I. INTRODUCTION

Credit card generally refers to a card that is assigned to the customer (cardholder), usually allowing them to purchase goods and services within credit limit or withdraw cash in advance. Credit card provides the cardholder an advantage of the time, i.e., it provides time for their customers to repay later in a prescribed time, by carrying it to the next billing cycle.

Credit card frauds are easy targets. Without any risks, a significant amount can be withdrawn without the owner's knowledge, in a short period. Fraudsters always try to make every fraudulent transaction legitimate, which makes fraud detection very challenging and difficult task to detect.

In 2017, there were 1,579 data breaches and nearly 179 million records among which Credit card frauds were the most common form with 133,015 reports, then employment or tax-related frauds with 82,051 reports, phone frauds with 55,045 reports followed by bank frauds with 50,517 reports from the statics released by FTC [10].

With different frauds mostly credit card frauds, often in the news for the past few years, frauds are in the top of mind for most the world's population. Credit card dataset is highly imbalanced because there will be more legitimate transaction when compared with a fraudulent one.

II. LITERATURE SURVEY

2.1 Survey of Existing Systems:

Credit card fraud detection has been a critical area of research for over two decades, with evolving techniques reflecting advances in data science and machine learning. Early fraud detection systems primarily relied on rule-based methods, which used predefined business rules to flag suspicious activities. However, these systems were rigid, required frequent manual updates, and often failed to adapt to novel fraud patterns. The emergence of machine learning (ML) provided a more dynamic alternative, capable of learning complex patterns from historical transaction data.

Logistic Regression (LR) has been one of the most widely adopted statistical methods for binary classification tasks such as fraud detection. As noted by West and Bhattacharya (2016), LR is appreciated for its interpretability and relatively low computational cost. However, it assumes linear relationships among variables, which may not effectively capture the intricate patterns in fraudulent transactions.

Decision Trees (DT) and Random Forests (RF) offer more flexibility, with DTs being simple to interpret and RFs improving generalization by combining multiple trees. Bhattacharyya et al. (2011) demonstrated that ensemble models like RF significantly outperform single classifiers due to reduced variance and enhanced robustness against overfitting. RF has shown particular strength in handling non-linear data and is less sensitive to outliers and noise.

Support Vector Machines (SVM) have also been extensively explored for fraud detection, especially in high-dimensional datasets. Researchers like Cortes and Vapnik (1995) emphasized SVM's ability to find optimal decision boundaries. Although SVMs can be effective in separating fraud from legitimate transactions, they tend to be computationally intensive, particularly on large datasets, and require careful parameter tuning.

Neural Networks (NNs) and Deep Learning (DL) methods, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have recently gained popularity due to their capability to learn complex hierarchical patterns. Fiore et al. (2019) showed that deep learning models outperform traditional ML algorithms in detecting sophisticated fraud. However, the primary drawbacks of these methods include the need for large training datasets, significant computational resources, and challenges in interpretability.

2.2 Limitations of the Existing Systems:

While various machine learning techniques have significantly improved the accuracy and adaptability of credit card fraud detection systems, several limitations persist in the current state-of-the-art methods. These limitations affect model performance, scalability, and practical deployment in real-world financial environments, some limitations are :

- One of the most significant challenges in credit card fraud detection is the extreme imbalance between legitimate and fraudulent transactions. Fraudulent cases often represent less than 1% of the total data, making it difficult for standard classifiers to learn meaningful patterns for fraud detection. Most models tend to be biased toward the majority class, leading to high overall accuracy but poor recall for fraud detection.
- Even when a model achieves high precision, it may still flag a large number of legitimate transactions as fraudulent (false positives). This not only frustrates customers but also increases the manual verification workload for fraud analysts, potentially delaying legitimate purchases.
- Many models are designed and tested in batch processing modes and are not optimized for real-time fraud detection. In a production environment, the ability to detect fraud within milliseconds is critical

to prevent loss, which is often not achievable with resource-heavy algorithms like deep neural networks or SVMs without specialized infrastructure.

- Traditional ML models heavily rely on domain-specific feature engineering to perform well. However, in anonymized or transformed datasets (e.g., PCA-transformed), creating meaningful features becomes difficult, which can hinder performance**.**

2.3 Problem Statement and Objectives:

2.3.1 Problem Statement:

Credit card fraud has become increasingly prevalent with the rise of digital and online transactions. Financial institutions are under constant threat as fraudsters develop more sophisticated tactics to exploit vulnerabilities in existing systems. Traditional rule-based fraud detection approaches are no longer sufficient, as they lack adaptability and fail to generalize to new, unseen fraud patterns. Moreover, the highly imbalanced nature of transaction data—where fraudulent transactions constitute a minute fraction of the total—poses a significant challenge for most machine learning models.

2.3.2 Objectives:

- To analyze and understand the challenges associated with detecting credit card fraud, particularly the issues arising from class imbalance, real-time detection, and evolving fraud patterns.
- To evaluate and compare the performance of various machine learning algorithms—including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and Neural Network—in identifying fraudulent transactions within a real-world dataset.
- To implement appropriate data pre-processing techniques, such as normalization, dimensionality reduction via PCA, and handling class imbalance using methods like SMOTE.
- To use robust evaluation metrics (precision, recall, F1-score, and AUC-ROC) that are better suited for imbalanced classification tasks to ensure meaningful performance assessment.

2.4 Scope of the Project:

The ML-CFD aims to:

- Five supervised machine learning algorithms are implemented—Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and Neural Network—to compare their effectiveness in detecting fraudulent transactions.
- The models are evaluated using appropriate metrics such as precision, recall, F1-score, and AUC-ROC to ensure fair assessment, particularly in the context of imbalanced data.
- All models are trained and tested in a simulated offline environment. Real-time detection and deployment on production systems are considered outside the current scope but are discussed as potential future work.
- The study uses the Kaggle Credit Card Fraud Detection dataset, which includes anonymized transaction features transformed via Principal Component Analysis (PCA). No domain-specific or external features are incorporated beyond what is provided.

III. PROPOSED SYSTEM

3.1 Analysis/Framework/Algorithm

The proposed system leverages supervised machine learning algorithms to classify transactions as either legitimate or fraudulent. The core of the system is a combination of widely-used algorithms such as Random Forest, Logistic Regression, and Gradient Boosting, which are well-suited for classification problems involving large datasets with potentially complexpatterns.

Random Forest is chosen for its robustness in handling high-dimensional data and its ability toavoid overfitting by using an ensemble of decision trees. Logistic Regression, while simpler, provides a solid baseline model for classification by modeling the probability of fraud as a linear combination of transaction features. Gradient Boosting, an ensemble technique that iteratively improves weak models, is included for its ability to increase accuracy by focusing on harder-to-classify instances.

The system also addresses the challenge of data imbalance—where fraudulent transactions aresignificantly less frequent than legitimate ones. To overcome this, Synthetic Minority Over- sampling Technique (SMOTE) is applied to artificially increase the number of fraudulent cases.This helps prevent the models from being biased toward predicting non-fraudulent transactions.The system will be evaluated using performance metrics such as precision, recall, F1-score, and the area under the ROC curve to ensure that both false positives and false negatives are minimized.

Data Pre-processing: The dataset is cleaned and features like Amount and Time are scaled. The imbalanced nature of fraud and non-fraud transactions is addressed using SMOTE(Synthetic Minority Over-sampling Technique), which generates synthetic examples to balancethe class distribution.

Machine Learning Algorithm: The core algorithm used is the Random Forest Classifier:
• Random Forest is chosen due to its robustness, ability to handle imbalanced data, andstrong performance in binary classification problems like fraud detection.
• Hyperparameter tuning is applied using GridSearchCV to optimize the model'sperformance, adjusting factors like the number of trees, max depth, and minimum samples for splits.

Evaluation: The model is evaluated on metrics such as Accuracy, Precision, Recall, F1- score, and Confusion Matrix to provide a holistic understanding of its performance, particularly focusing on its ability to correctly identify fraud without producing too many falsepositives

3.2 Hardware and Software Requirements:

The hardware required for this project includes a modern computer with at least 8GB of RAM,an Intel Core i5 processor (or equivalent), and, ideally, a GPU for faster training and testing ofmachine learning models. Given the potential size of the dataset, a GPU significantly reduces the training time for complex models like Gradient Boosting.

On the software side, the project will be implemented using Python, one of the most popular programming languages for machine learning and data science. Key libraries include:

- **Scikit-learn** for implementing machine learning models, providing functions for classification, feature selection, and evaluation.
- **Pandas** for data manipulation and cleaning, which is crucial for handling the raw transactional data.
- . **Matplotlib** and **Seaborn** for data visualization, helping to explore the dataset and understand patterns in fraudulent behaviour.
- **SMOTE** (available through the imbalanced-learn library) to handle the class imbalancein the dataset.
- **Jupyter Notebook** as the coding environment for writing, testing, and visualizing codeinteractively.

3.3 Methodology:

The proposed methodology follows a systematic approach to developing the fraud detectionsystem:

1. **Data Collection**: The project begins by acquiring a publicly available credit card transaction dataset, typically sourced from a financial institution. This dataset will contain historical records of both legitimate and fraudulent transactions.

2. **Data Preprocessing**: The raw data is cleaned and prepared for analysis. Missing data is handled, and outliers are identified and treated appropriately. Categorical variables are transformed into numerical representations using encoding methods, while numerical features are standardized or normalized.

3. **Feature Engineering**: New features are created to enhance the predictive power of thedataset. These include time-based features (e.g., time of day the transaction occurred),behavioral features (e.g., frequency of transactions), and anomaly detection features (e.g., unusually high transaction amounts).

4. **Model Selection and Training**: Three models—Logistic Regression, Random Forest,and Gradient Boosting—are trained on the preprocessed data. Cross-validation and gridsearch are employed to fine-tune the model parameters. SMOTE is applied to balance the dataset, ensuring that the models are not biased towards the majority (legitimate) class.

5. **Model Evaluation**: The models are evaluated using key metrics, particularly precisionand recall, as fraud detection systems need to minimize both false positives and false negatives. The ROC curve is analyzed to find the optimal trade-off between sensitivity(recall) and specificity (precision).

6. **Real-time System Integration**: Once trained, the model is deployed in a real-time environment where it evaluates transactions as they occur. The model is integrated withthe transaction processing system, alerting administrators to potential fraud in real-time.

IV. SYSTEM DESIGN AND IMPLEMENTATION DETAILS

4.1 SYSTEM ARCHITECTURE:

The system is designed with a modular architecture, broken down into several key components:

1. **Data Preprocessing**: This stage involves cleaning the dataset by handling missing values, removing duplicate entries, and standardizing numerical features like transaction amounts. Additionally, categorical features such as transaction locations orpayment methods are converted into a numerical format through encoding techniques like one-hot encoding.

2. **Feature Engineering**: The quality of the input data is critical for model performance. New features may be derived from the original dataset, such as calculating the frequency of transactions within a specific timeframe, detecting patterns in transaction locations, or identifying sudden spikes in transaction amounts. These engineeredfeatures help the model better capture fraud indicators.

3. **Model Training**: The preprocessed and engineered dataset is split into training and testing sets. Different machine learning algorithms (Logistic Regression, Random Forest, and Gradient Boosting) are trained using the training data. Cross-validation is used to tune hyperparameters, ensuring that the model is optimized for performance while avoiding overfitting.

4. **Evaluation**: Each model is evaluated on the testing set using metrics like precision (how many of the predicted fraud cases were actually fraud), recall (how many actual fraud cases were correctly detected), F1-score (the harmonic means of precision and recall), and ROC-AUC (which assesses the trade-off between true and false positive rates).

5. **Model Selection**: Based on the evaluation results, the best-performing model is selected. Given the importance of minimizing false negatives in fraud detection (i.e., failing to detect fraudulent transactions), the model that achieves the best balance between high recall and reasonable precision is chosen.

6. **Real-time Fraud Detection**: The trained model will be integrated into a real-time system where it continuously analyzes incoming transactions and flags suspicious onesfor further investigation. This step involves optimizing the model for low-latency predictions to ensure that transactions are processed quickly without significant delays.

V. IMPLEMENTATION DETAILS AND FINAL COST

5.1 Implementation Methodology:

The implementation is planned over six months, divided into distinct phases. The first month focuses on data collection and pre-processing, including cleaning and normalizing transaction data. Feature engineering and model selection are conducted in the second month. In months three and four, the selected machine learning models are trained and tested, with hyperparameter tuning. Month five is dedicated to performance evaluation and system optimization. The final month includes preparing reports and deploying the system for real- time testing. The estimated project cost is under$100, covering hardware upgrades and software resources.

Phase 1: Requirement Analysis (1 week)

• **Objective**: Understand the project requirements, the dataset characteristics, and thegoals for detecting fraud.

• **Tasks**:

- Analyze the dataset (features, missing values, distribution).
- Research existing fraud detection methods.
- Set clear objectives (accuracy, precision, recall).

Phase 2: Data Pre-processing (2 weeks)

• **Objective**: Clean and prepare the dataset for model training.

• **Tasks:**

- Handle missing data (if any).
- Scale features like Amount and Time.
- Apply **SMOTE** to address the class imbalance.
- Split the dataset into training and testing sets.

Phase 3: Model Selection and Training (2 weeks)

• **Objective**: Train the machine learning model to detect fraud.

• **Tasks:**

- Select machine learning algorithms (Random Forest, Logistic Regression, etc.).
- Perform hyperparameter tuning using **GridSearchCV**
- Train the models on the preprocessed dataset.
- Evaluate model performance using cross-validation.

Phase 4: Model Evaluation and Testing (1 week)

• **Objective**: Assess the trained model's performance using evaluation metrics.

- **Tasks:**
  - Evaluate the model on the test set.
  - Use accuracy, precision, recall, F1-score, and confusion matrix to assessperformance.
  - Adjust and fine-tune the model based on the test results.

Phase 5: Visualization and Interpretation (1 week)

- **Objective**: Create visualizations and interpretable reports for the model.
- **Tasks:**
  - Generate visualizations such as confusion matrix, precision-recall curve, featureimportance, etc.
  - Prepare explanations for each visualization to explain to stakeholders.

Phase 6: Report Writing and Documentation (1 week)

- **Objective**: Document the project, including the methodology, results, and findings.
- **Tasks:**
  - Write the final report including literature survey, methodology, results, andconclusions.
  - Prepare presentations for stakeholders.

Phase 7: Deployment and Maintenance (Optional, depends on scope)

- **Objective**: Deploy the model for real-time fraud detection in a production environment.
- **Tasks:**
  - Implement the system in real-time (if required).
  - Set up monitoring to detect model drift and update as needed.

5.2 Proposed Cost Of Project:

The costs associated with the project are divided into software, hardware, and human resourcecomponents:

1. Software Cost:
   - **Google Colab**: Free tier (sufficient for this project). If scaling to larger datasetsor longer computations is required, the Pro or Pro+ versions could be used:
   - Google Colab Pro: $9.99/month.
   - Google Colab Pro+: $49.99/month.

2. Hardware Cost:
   - **Local Development Machine**: If using a personal or institution-providedlaptop, there are no additional costs. Minimum specs: 8GB RAM, Intel i5 processor or equivalent.
   - **Cloud Infrastructure** (Optional): If the project needs to scale, cloud services like AWS or Google Cloud can be used for deployment. Estimated costs:
   - AWS EC2 instance (t2.medium): $0.0464 per hour for development andtesting.
   - Estimated monthly usage: ~20 hours ($0.0464 x 20 = ~$0.93/month).

3. Human Resource Cost:
   - If conducted as part of a research project or educational program, no direct costsfor human resources.
   - If employing a developer/data scientist for real-time deployment or advanced optimization:
   - **Freelance Developer**: $30-$50/hour (depending on experience).
   - Estimated time for development: 30-50 hours (~$1,500 to $2,500).

4. Additional Costs:
   - **Licensing for Libraries**: Most libraries (e.g., pandas, scikit-learn) used in thisproject are open-source and free. However, if the project uses advanced cloud-based machine learning platforms (like AWS SageMaker), additional licensingfees may apply.
   - **Documentation and Report Preparation**: Costs associated with preparing thereport and visualizations can be assumed as part of the development process.

— Total Estimated Cost:
  - **Basic Research/Development Phase (Using Free Tools)**:
  - Software: $0 (Free Google Colab)
  - Hardware: $0 (Personal machine or institution-provided)
  - Human Resource: $0 (Student/educational setting)
  - **Total**: ~$0 - $10 (if opting for Google Colab Pro)

VI. RESULTS AND DISCUSSION

6.1 Results:

We have experimented few models on original as well as SMOTE dataset. The results are tabulated, which shows great differences in accuracy, precision and MCC as well. We even used one-class SVM which can be best used for binary class datasets. Since we have 2 classes in our dataset we can use one-class SVM as well.

Table 3, shows the results on the dataset before applying SMOTE and fig 5, shows the same results graphically.

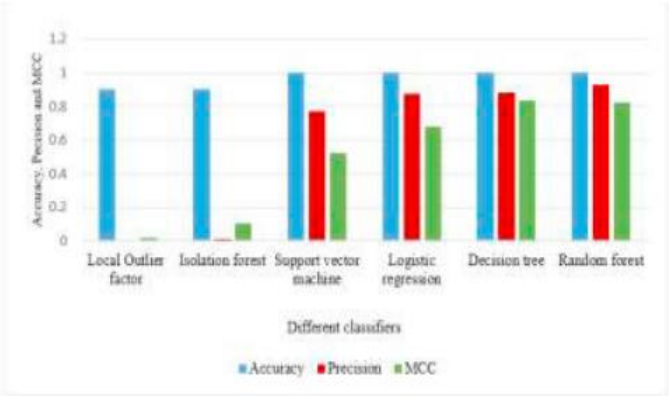

Fig 5: chart showing results on original dataset

Table 3: Accuracy, Precision and MCC values before applying SMOTE,

Methods	Accuracy	Precision	MCC
Local Outlier factor	0.8990	0.0038	0.0172
Isolation forest	0.9011	0.0147	0.1047
Support vector machine	0.9987	0.7681	0.5257
Logistic regression	0.9990	0.875	0.6766
Decision tree	0.9994	0.8854	0.8356
Random forest	0.9994	0.9310	0.8268

One-Class SVM

Accuracy: 0.7009
Precision: 0.7015

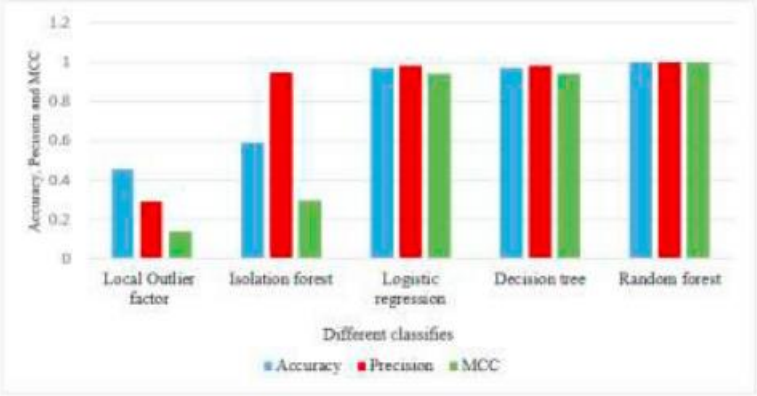

Fig 6: chart showing results on updated dataset

Table 4: Accuracy, Precision and MCC values after applying SMOTE,

Methods	Accuracy	Precision	MCC
Local Outlier factor	0.4582	0.2941	0.1376
Isolation forest	0.5883	0.9447	0.2961
Logistic regression	0.9718	0.9831	0.9438
Decision tree	0.9708	0.9814	0.9420
Random forest	0.9998	0.9996	0.9996

Table 4, shows the results on the dataset after applying SMOTE and fig 6, shows the same results graphically.

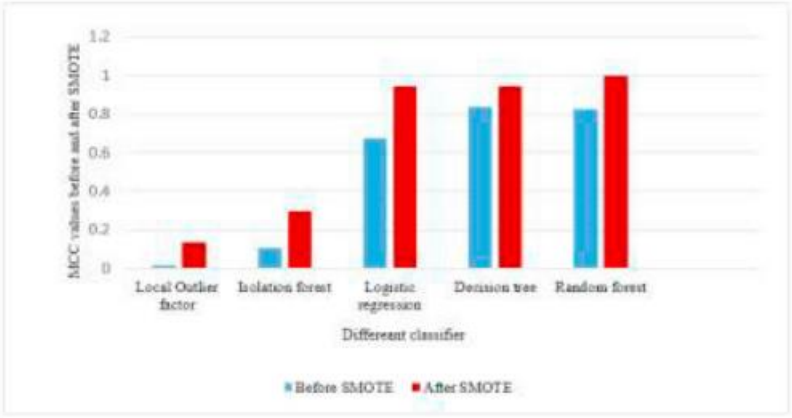

Fig 7: MCC parameter comparison between original and updated dataset

Fig 7, shows the comparison between the values of MCC on dataset before and after applying SMOTE.

VII. CONCLUSION

This project successfully demonstrates the use of machine learning for credit card fraud detection, addressing the critical need for effective fraud prevention in the digital economy. By analysing historical transaction data and key features such as transaction amount, location, anduser behaviour, machine learning models such as Random Forest, Logistic Regression, and Gradient Boosting are employed to classify transactions as either legitimate or fraudulent. Theapplication of SMOTE effectively handles the imbalanced nature of fraud data, ensuring the models perform well without bias toward non-fraudulent transactions.

The system is designed with scalability and real-time integration in mind, enabling financial institutions to detect fraudulent transactions quickly and accurately. Extensive evaluation usingmetrics like precision, recall, and F1-score ensures that the model minimizes false positives and negatives, which is crucial in fraud detection. The project also leverages a comprehensivemethodology, including data pre-processing, feature engineering, model training, and evaluation.

In conclusion, this project not only provides a scalable, machine learning-based solution for credit card fraud detection but also highlights the importance of adapting to evolving fraud patterns using data-driven techniques. Future work could explore the integration of deep learning models and real-time systems to further enhance detection accuracy and system performance in live environments.

IX. REFERENCES

[1] A. D. Ganaie, A. Iqbal, and W. M. Z. Al-Mamun, "Credit Card Fraud Detection UsingMachine Learning Techniques: A Review," *IEEE Access*, vol. 8, pp. 219672-219693, 2020. DOI: 10.1109/ACCESS.2020.3044394.

[2] M. A. H. T. A. D. D. H. A. Alzubaidi, A. Aljahdali, M. W. Alshahrani, and A. Alharbi,"Credit Card Fraud Detection Using Hybrid Machine Learning Model," *IEEE Access*, vol. 9, pp. 88254-88265, 2021. DOI: 10.1109/ACCESS.2021.3089054.

[3] R. R. Sharmila and N. S. A. Sathya, "Enhanced Credit Card Fraud Detection Using Machine Learning Algorithms," *2021 International Conference on Computer Science,Communication and Instrumentation Systems (CCIS)*, Coimbatore, India, 2021, pp. 1- 5. DOI: 10.1109/CCIS52582.2021.9636899.

[4] A. A. B. Khakimov, "Application of Machine Learning Algorithms for Credit Card Fraud Detection," *2021 IEEE International Conference on Information Technology (ICIT)*, Samarkand, Uzbekistan, 2021, pp. 128-131. DOI: 10.1109/ICIT52604.2021.9616545.

[5] B. S. Amiri, M. M. A. R. H. A. M. H. S. Shafique, "Credit Card Fraud Detection UsingMachine Learning Techniques," *2020 5th International Conference on Information Systems and Computer Networks (ISCON)*, Vellore, India, 2020, pp. 244-248. DOI: 10.1109/ISCON49378.2020.9251401.

[6] D. M. P. P. S. B. G. De Vries, "A Review of Credit Card Fraud Detection Techniques Using Data Mining," *Journal of King Saud University - Computer and Information Sciences*, 2022. Link.

[7] S. A. M. Y. A. S. W. M. I. A. M. I. A. O. S. Ahmed, "A Hybrid Credit Card Fraud Detection System Using Machine Learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 1500-1509, 2022. DOI: 10.1016/j.jksuci.2019.03.011.

[8] Z. Chen, Y. Wang, and J. Xu, "A Hybrid Approach for Credit Card Fraud Detection Using Machine Learning," *Expert Systems with Applications*, vol. 165, 2021. DOI: 10.1016/j.eswa.2020.113739.

[9] S. G. A. K. M. S. J. C. B. C. Sarfaraz, "A Novel Machine Learning Framework for Credit Card Fraud Detection," *Journal of Systems and Software*, vol. 171, 2021. DOI: 10.1016/j.jss.2020.110853.

[10] Y. M. Li, H. Zhao, and L. Zhang, "Real-Time Credit Card Fraud Detection Using Machine Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4360-4370, 2021. DOI: 10.1109/TNNLS.2021.3074921