Credit Card Fraud Detection

Lokesh Saroj, Prof. Sumona Mondal, Naveen Reddy
Department of Data Science
Clarkson University

Abstract—Credit card fraud poses significant challenges to financial institutions, exacerbated by the increasing reliance on digital transactions and evolving fraudulent techniques. This project focuses on developing a machine learning-based system to detect fraudulent transactions accurately while addressing key challenges, such as class imbalance and feature optimization. Using the Credit Card Fraud Detection Dataset, which includes only 0.39% fraudulent transactions, the study emphasizes data preprocessing, feature engineering, and advanced modeling. Data preprocessing involved normalizing numerical features, encoding categorical variables, and creating new features, such as transaction time and customer demographics. Statistical tests, including ANOVA and Chi-Square, identified key predictors like transaction amount, hour, and category. To address class imbalance, the Synthetic Minority Oversampling Technique (SMOTE) was applied, with 70% oversampling achieving a balance between recall and precision. Logistic Regression and Random Forest models were implemented and compared. Random Forest with SMOTE outperformed other approaches, achieving a recall of 0.75 while identifying significant predictors and actionable fraud patterns. Logistic Regression, while interpretable, struggled with imbalanced data. This study highlights the effectiveness of machine learning in fraud detection and provides a foundation for future enhancements, such as advanced models, real-time detection, and continuous learning, to improve scalability and precision in real-world applications.

1 Introduction

In today's digital age, the increased reliance on credit cards for online and offline transactions has made financial systems more susceptible to fraudulent activities. Credit card fraud has become a serious issue for financial institutions, costing billions of dollars annually and leading to compromised security for customers. Traditional fraud detection methods, though effective, often struggle to keep up with the rapidly evolving techniques used by fraudsters. As a result, there is a growing need for more advanced, intelligent systems to identify fraudulent transactions in real time. This project focuses on applying machine learning techniques to detect fraudulent credit card transactions. Machine learning can learn from historical transaction data and distinguish between legitimate and suspicious patterns. By analyzing large datasets containing past transaction records, machine learning models can help identify anomalies and predict fraudulent behavior more accurately and efficiently than traditional rule-based systems. The objective of this project is to build a robust fraud detection system using a variety of machine learning algorithms, such a logistic regression, decision trees, and random forests. By comparing the performance of different models, we aim to develop a solution that minimizes false positives and maximizes the detection of fraudulent activities. The project will explore key features of credit card transactions, such as transaction amount, location, and time, and use these to train and evaluate the models. In this report, we will provide a detailed overview of the machine learning algorithms used, the data preprocessing steps taken, and the results obtained from applying these models. Additionally, we will discuss the challenges associated with fraud detection, such as handling imbalanced datasets, and propose strategies to address these issues.

2 OBJECTIVE

The objective of this project is to develop a comprehensive credit card fraud detection system that leverages data mining and machine learning techniques to accurately identify fraudulent transactions. The project focuses on addressing key challenges in fraud detection while ensuring practical applicability in real-world scenarios. Below are the detailed goals of the project:

2.1 Accurate Fraud Detection

Fraudulent transactions often account for a small percentage of the overall dataset, making it crucial to develop models that can identify these cases with high precision and recall. The aim is to minimize false negatives (missed frauds) while maintaining acceptable levels of false positives.

2.2 Handling Class Imbalance

The dataset exhibits a significant imbalance, with fraudulent cases comprising only 0.39 percent of the total data. Addressing this imbalance is critical to ensuring that machine learning models do not bias predictions toward the majority class (non-fraud). Techniques such as SMOTE (Synthetic Minority Oversampling Technique) are employed to generate synthetic samples for the minority class, enabling balanced training and improved model generalization.

2.3 Feature Engineering

Feature engineering is a key component of this project to enhance the dataset's predictive power. New features such as transaction time patterns transaction hour and transaction weekday, geographic regions, and customer age bands are derived to improve model performance and interpretability. These features help uncover hidden patterns in fraudulent transactions.

2.4 Model Evaluation and Selection

Multiple machine learning models, including Logistic Regression and Random Forest, are implemented and evaluated. The goal is to compare these models based on key metrics such as precision, recall, F1-score, and AUC-ROC to determine the best-performing approach. The focus is on selecting a model that balances detecting fraudulent transactions and minimizing false positives.

2.5 Operational Insights

The project aims to provide actionable insights into fraud detection trends. For instance, understanding temporal patterns (e.g., high fraud rates during specific hours) and geographic hotspots helps organizations target preventive measures effectively. These insights will assist financial institutions in strengthening their fraud prevention frameworks.

2.6 Real-World Applicability

The ultimate objective is to build a fraud detection system that is deployable in real-time environments. The model should be capable of handling large volumes of transactions while delivering reliable predictions. This ensures financial institutions can use the system to monitor and mitigate risks efficiently.

By addressing these objectives, this project not only focuses on improving the accuracy of fraud detection but also ensures practical usability and applicability in real-world scenarios.

3 DATASET OVERVIEW

The Credit card fraud detection dataset sourced from Kaggle consists of 555,719 rows and 22 columns, with only 0.39% of the transactions labeled as fraudulent, highlighting a significant class imbalance. The dataset includes detailed transactional information such as transaction date and time (trans_date_trans_time), amount (amt), merchant details (merchant), and transaction categories (category). It also provides customer-specific details like anonymized credit card numbers (cc_num), gender, location

data (city, state, zip), and demographic information such as date of birth (dob) and occupation (job). Geographic variables such as latitude and longitude for both customers (lat, long) and merchants (merch_lat, merch_long) offer spatial insights into transaction patterns. Additionally, the dataset includes population details for customer cities (city_pop) and timestamps in Unix format (unix_time). The target variable, is_fraud, identifies whether a transaction is fraudulent (1) or not (0). This comprehensive dataset serves as a robust foundation for fraud detection through machine learning and feature analysis.

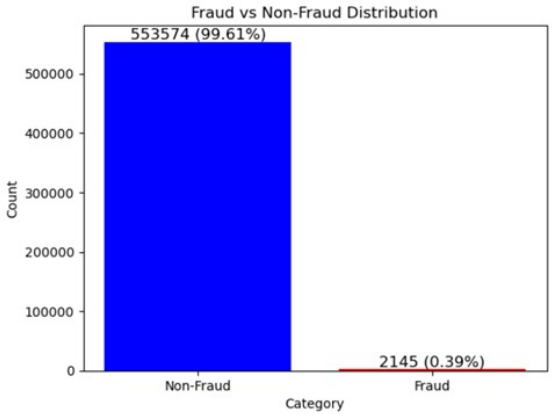

Fig. 1: Data Distribution information

4 METHODS

The methodology for this project was designed to systematically address the challenges posed by the imbalanced nature of the dataset and ensure robust fraud detection through effective preprocessing, feature engineering, and model evaluation. The project workflow included six primary steps:

4.1 Data Preprocessing

The raw dataset underwent extensive cleaning to ensure its readiness for analysis. Initially, all columns were retained, as the focus was on understanding the dataset comprehensively. During feature engineering and after conducting statistical tests, irrelevant columns such as street, dob, and city were identified and removed, as they contributed little predictive value or contained excessive noise. Missing values, if present, were either imputed using median values or removed to maintain dataset integrity. Numerical features, particularly amt (transaction amount), were normalized using robust scaling to handle outliers, ensuring features were on comparable scales. Categorical variables like category and gender were transformed using one-hot encoding and label encoding, enabling compatibility with machine learning models.

4.2 Feature Engineering

The dataset underwent several feature engineering processes to enhance its predictive capabilities and uncover patterns indicative of fraudulent transactions. Temporal features such as transaction hour and transaction weekday were extracted from the trans_date_trans_time column to reveal time-based trends. This allowed for the identification of peak fraud periods during specific hours or days of the week. Geographic data, including customer and merchant latitude (lat) and longitude (long), were analyzed to detect anomalies in transaction locations. Additionally, states were grouped into broader geographic regions (e.g., 'Rocky Mountains', 'Far West', 'South',' Midwest', 'Northeast') to simplify analysis and identify regional fraud trends. Demographic features were also refined by categorizing customer ages into predefined groups such as "18-35," "36-60," and "60+." This facilitated demographic analysis of fraud susceptibility. Binary demographic variables, such as gender and is_male, were encoded to ensure compatibility with machine learning models. Behavioral features were created by calculating metrics like transaction frequency and average transaction amount per user, which helped establish baseline spending patterns. Significant deviations from these baselines were flagged as potential indicators of fraud. Feature encoding techniques, including one-hot encoding for nominal categorical variables like category and label encoding for ordinal variables, were applied to prepare the data for machine learning models. The target variable.

4.3 ANOVA Test for Numerical Variables

The Analysis of Variance (ANOVA) test was conducted to evaluate the statistical significance of numerical features in distinguishing between fraudulent (is_fraud = 1) and non-fraudulent (is_fraud = 0) transactions. This test compared the variance within groups to the variance between groups, with a p-value less than 0.05 indicating a significant relationship. The results showed that amt (transaction amount) had the highest F-statistic (5430.46) and a p-value of 0.00, confirming it as a critical predictor. Similarly, transact_hour (hour of transaction) and merch_lat (merchant latitude) were significant with p-values of 0.00. However, merch_long (merchant longitude) and distance_km (distance between customer and merchant) had high p-values (0.40 and 0.90, respectively), indicating they were not significant and were dropped. The ANOVA test refined the feature set by retaining only the significant numerical variables, improving the model's predictive power and eliminating irrelevant features.

Variable	F-statistic	p-value	Finding
amt	5430.46	0	Significant p-value < α (0.05)
unix_time	94.89	0	
lat	19.1	0	
merch_lat	18.77	0	
city_pop	13.4	0	
merch_long	0.62	0.43	Not significant p-value ≥ α (0.05)
long	0.52	0.47	
distance_km	0.03	0.86	

Fig. 2: ANOVA Test Results for Feature Significance Analysis. The table highlights the F-statistic, p-value, and the decision to retain or drop features based on their significance in predicting fraudulent transactions.

4.4 Chi-Square Test on Categorical Variables

The Chi-Square Test was conducted to evaluate the statistical significance of categorical variables in distinguishing fraudulent (is_fraud = 1) and non-fraudulent (is_fraud = 0) transactions. This test measures the independence between a categorical variable and the target variable by comparing observed and expected frequencies. A p-value less than 0.05 indicates a significant relationship between the variable and fraud detection. The results showed that variables such as category_shopping_net, category_grocery_pos, and transaction_weekday had high Chi-squared scores and p-values of 0.00, confirming their strong predictive relationship with fraud. Similarly, variables like category_food_dining and category_health_fitness were also significant and retained for further analysis. However, variables such as region, age_group, and gender_M had p-values greater than 0.05, indicating no significant relationship, and were excluded. The Chi-Square test refined the categorical feature set by identifying key predictors of fraud while removing less relevant variables, enhancing the efficiency of the modeling process.

Feature	Chi2 Score	p-value	Finding
category_shopping_net	739.83	0	Significant **p-value < α (0.05)**
category_grocery_pos	393.98	0	
category_misc_net	247.46	0	
category_home	90.61	0	
transaction_weekday	84.84	0	
category_kids_pets	80.73	0	
category_food_dining	63.05	0	
category_health_fitness	56.88	0	
category_personal_care	44.25	0	
category_misc_pos	28.41	0	
category_gas_transport	18.65	0	
category_grocery_net	15.46	0	
category_travel	11.15	0	
Region	8.77	0	
age_group	2.73	0.1	Not significant **p-value ≥ α (0.05)**
category_shopping_pos	2.26	0.14	
gender_M	0.17	0.68	

Fig. 3: Chi-Squared Test Results for Categorical Variables. This table summarizes the Chi2 scores, p-values, and findings for various categorical features, identifying significant predictors to retain and insignificant predictors to drop.

4.5 Addressing Class Imbalance

The dataset exhibited a significant class imbalance, with fraudulent transactions comprising

only 0.39% of the total data. This posed a challenge for machine learning models, which tend to favor the majority class (non-fraud) in such scenarios, leading to poor detection of fraudulent cases. To address this issue, the Synthetic Minority Oversampling Technique (SMOTE) was implemented. SMOTE works by generating synthetic samples for the minority class (is_fraud = 1) rather than duplicating existing ones. This was achieved by creating synthetic data points along the line segments joining minority class samples within feature space. To identify the optimal level of oversampling, SMOTE was tested with 30%, 50%, 70%, and 100% sampling strategies. Based on the evaluation of model outputs for each strategy, it was observed that 70% oversampling provided the best balance between recall and precision, significantly improving the models' ability to detect fraudulent transactions while minimizing false positives. This balanced ratio ensured sufficient representation of the minority class in the training data without introducing unnecessary noise or overfitting. By selecting the 70% sampling strategy, the machine learning algorithms were able to learn more effectively, distinguishing fraudulent transactions without bias toward the majority class. The implementation of SMOTE with this optimized sampling ratio not only enhanced the models' recall for fraud detection but also ensured robust performance on unseen data, addressing the class imbalance challenge comprehensively.

4.6 Model Selection

Two machine learning models, Logistic Regression and Random Forest, were implemented and compared. Logistic Regression was used as a baseline model, while Random Forest, an ensemble technique, was selected for its ability to handle non-linear relationships and feature importance analysis.

5 Model Evaluation

To detect fraudulent transactions effectively, two machine learning models were utilized: Logistic Regression and Random Forest. Each model was evaluated both with and without the application of SMOTE to address the dataset's significant class imbalance. Below is a detailed explanation of these models and their functionalities.

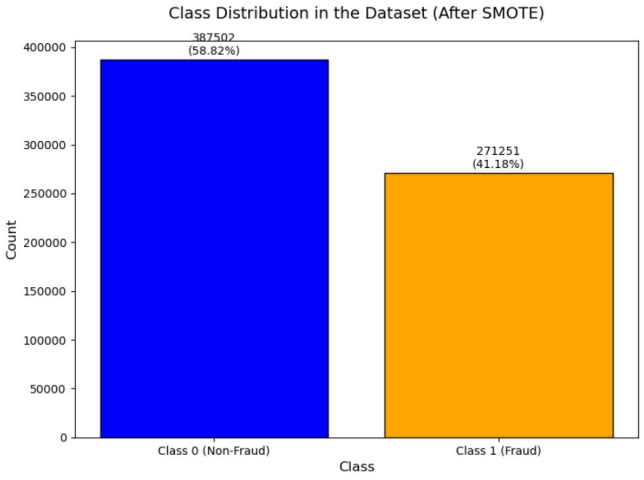

Fig. 4: Class distribution in the dataset after applying SMOTE. The synthetic minority oversampling technique (SMOTE) was used to balance the dataset, resulting in 58.82% non-fraud (Class 0) and 41.18% fraud (Class 1). This adjustment addresses the class imbalance issue, improving model performance.

5.1 Logistic Regression model

Logistic Regression is a statistical model used for binary classification tasks. It works by estimating the probability of an event occurring (e.g., a transaction being fraudulent) using a logistic function, which maps the outputs to a range between 0 and 1. The model calculates a linear combination of input features, applies a logistic transformation, and classifies the output into one of two categories (is_fraud = 1 or is_fraud = 0)

How it Performs: Without SMOTE, the Logistic Regression model prioritized precision, correctly identifying non-fraudulent transactions while struggling to recall fraudulent ones due to the class imbalance. This behavior is common for models that are biased toward the majority class. With SMOTE, the model's recall improved significantly, as the oversampling technique provided it with more minority

class examples during training. However, this improvement came at the cost of slightly lower precision, as the model occasionally misclassified non-fraudulent transactions as fraudulent. Logistic Regression's simplicity and interpretability make it a reliable baseline model.

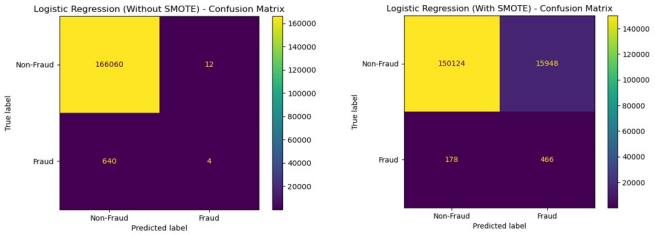

Fig. 5: Comparison of confusion matrices for Logistic Regression before and after applying SMOTE. The left matrix shows results without SMOTE, where the model has high precision but low recall for fraud detection. The right matrix shows results with SMOTE, where the recall for fraud improves significantly, although at the cost of higher false positives for non-fraud transactions.

5.2 Random Forest model

Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their outputs for prediction. Each tree in the forest is trained on a random subset of the data, and the final prediction is determined either by the majority class (classification) or the average output (regression). This approach helps reduce overfitting and improves the model's robustness and accuracy.

How it Performs:Without SMOTE, Random Forest effectively leveraged its ensemble structure to identify patterns in the dataset. While it performed well in terms of precision, the class imbalance limited its recall, causing it to miss some fraudulent transactions. With SMOTE, the model's recall improved significantly, as the balanced dataset allowed it to learn more about fraudulent patterns. This adjustment resulted in a well-rounded performance with high F1-scores. Random Forest also provided insights into feature importance, identifying amt (transaction amount) and transact_hour (hour of transaction) as key predictors, demonstrating its ability to handle complex relationships in the data.

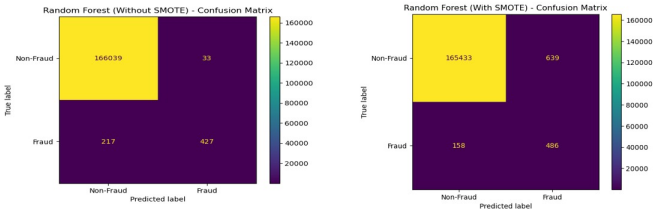

Fig. 6: Comparison of confusion matrices for Random Forest before and after applying SMOTE. The left matrix shows results without SMOTE, where the model has better precision but lower recall for fraud detection. The right matrix shows results with SMOTE, where recall for fraud improves significantly, reducing missed fraudulent transactions, albeit with a slight increase in false positives for non-fraudulent cases.

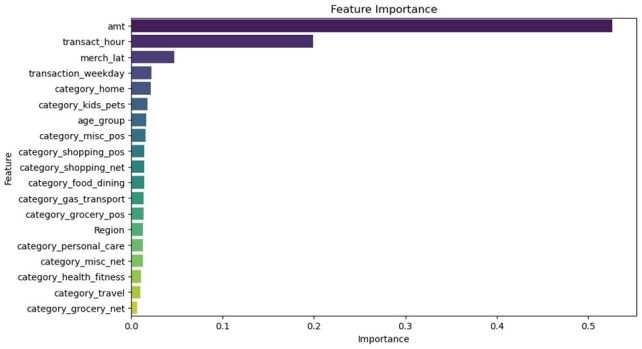

Fig. 7: Features Importance of Random Forest Model

6 Conclusion

The project aimed to tackle the critical challenge of credit card fraud detection by leveraging advanced machine learning techniques and addressing key issues like class imbalance and feature optimization. Through systematic preprocessing, feature engineering, and model evaluation, a robust fraud detection system was developed to identify fraudulent transactions effectively. The evaluation of machine learning models for credit card fraud detection identified Random Forest with SMOTE as the most

effective approach, achieving a high recall of 0.75. This performance aligns with the primary objective of fraud detection—minimizing false negatives and ensuring that a significant portion of fraudulent transactions is identified. However, this improvement in recall comes with a trade-off, as the model's lower precision (0.43) results in an increased rate of false positives. Conversely, Random Forest without SMOTE offers better precision but struggles to detect fraud effectively due to lower recall, highlighting the challenges in balancing these metrics. The Logistic Regression model, though interpretable and straightforward, struggled to handle the complexities of imbalanced data. While SMOTE significantly improved its recall, the model remained less effective overall when compared to Random Forest. Key fraud predictors, including transaction amount, time, and weekday, provided valuable insights into fraudulent patterns, such as peak hours and regional hotspots. These findings enable actionable strategies for fraud prevention, assisting financial institutions in targeting vulnerabilities more effectively. The implementation of SMOTE also proved critical in addressing class imbalance, with a 70% oversampling strategy optimizing the trade-off between recall and precision. In conclusion, this project establishes a strong foundation for fraud detection using machine learning. However, to build a truly efficient and reliable system, further exploration and refinement of techniques are essential to enhance the model's performance for real-world applications.

Model	Logistic Regression (Without SMOTE)	Logistic Regression (With SMOTE)	Random Forest (Without SMOTE)	Random Forest (With SMOTE)
Precision (Class 1)	0.25	0.03	0.93	0.43
Recall (Class 1)	0.01	0.72	0.66	0.75
F1-Score (Class 1)	0.01	0.05	0.77	0.55
Accuracy	1	0.9	1	1
Observations	High accuracy due to class imbalance but fails to detect fraud cases effectively (low recall).	Significant improvement in recall but low precision, leading to more false positives.	High precision and reasonable recall, better than Logistic Regression without SMOTE.	Improved recall compared to without SMOTE, but precision drops, leading to more false positives.

Fig. 8: Comparison of confusion matrices for Logistic Regression before and after applying SMOTE. The left matrix shows results without SMOTE, demonstrating high precision but low recall for fraud detection. The right matrix shows results with SMOTE, where recall improves significantly, capturing more fraudulent transactions but increasing false positives.

7 Future Scope

To enhance the credit card fraud detection system, several advanced techniques can be explored:

Advanced Machine Learning Models: Use algorithms like Gradient Boosting (GBM), XGBoost, and Deep Learning (e.g., Neural Networks, LSTMs) to capture complex and evolving fraud patterns.

Hybrid and Ensemble Models: Combine multiple models, such as Random Forest with boosting techniques, to leverage the strengths of each for improved precision and recall.

Anomaly Detection: Implement unsupervised techniques like Autoencoders, Isolation Forests, or One-Class SVM to identify fraudulent transactions without relying on extensive labeled data.

Improved Feature Engineering: Incorporate dynamic and temporal features, such as transaction trends over time or time-series modeling, to better capture fraud behavior.

Class Imbalance Solutions: Test alternative methods like ADASYN, Tomek Links, or cost-sensitive learning to handle the imbalance effectively while maintaining model accuracy.

Hyperparameter Optimization: Use automated tools like Grid Search or Bayesian Optimization to fine-tune model parameters for better performance.

Explainability: Apply tools such as SHAP or LIME to enhance model interpretability, making results more actionable for stakeholders.

Real-Time Fraud Detection: Develop a real-time detection system using tools like Apache Kafka or Spark Streaming to monitor transactions as they occur.

Data Augmentation: Use synthetic data generation to simulate realistic fraud scenarios and expand the training dataset.

Continuous Learning: Implement periodic

retraining of models with new transaction data to adapt to evolving fraud patterns.

Integration of External Data: Enrich the dataset with external information, such as demographic, geolocation, or merchant-specific data, to improve predictive accuracy.

By incorporating these strategies, the system can achieve better precision, scalability, and adaptability, making it more effective for real-world fraud detection scenarios.

8 Acknowledgments

I would like to extend my heartfelt gratitude to Prof. Sumona Mondal and Naveen Reddy for their invaluable guidance, support, and encouragement throughout this project. Their expertise, thoughtful insights, and constructive feedback have played a crucial role in shaping the direction and success of my work. Their dedication to fostering a supportive and intellectually stimulating environment has been a constant source of motivation, helping me navigate challenges and strive for excellence. I am deeply appreciative of the time and effort they have invested in mentoring me and sharing their knowledge, which has greatly enriched my academic journey. It has been an absolute privilege to learn under their mentorship, and I am sincerely grateful for their unwavering support and commitment. Their contributions have not only been instrumental in the completion of this project but have also left a lasting impact on my learning and growth.

9 References

Dighe, D., Patil, S., & Kokate, S. (2018). Detection of credit card fraud transactions using machine learning algorithms and Neural Networks: A comparative study. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). https://doi.org/10.1109/iccubea.2018.8697799

Domínguez-Almendros, S., Benítez-Parejo, N., & Gonzalez-Ramirez, A. R. (2011). Logistic regression models. Allergologia et immunopathologia, 39(5), 295-305.

Gupta, A., Lohani, M. C., & Manchanda, M. (2021). Financial fraud detection using naive Bayes algorithm in highly imbalance data set. Journal of Discrete Mathematical Sciences and Cryptography, 24(5), 1559–1572. https://doi.org/10.1080/09720529.2021.1969733

Itoo, F., Meenakshi, & Singh, S. (2020). Comparison and analysis of logistic regression, Naïve Bayes and Knn Machine Learning Algorithms for credit card fraud detection. International Journal of Information Technology, 13(4), 1503–1511. https://doi.org/10.1007/s41870-020-00430-y

Mahesh, B. (2020). Machine Learning Algorithms - A Review, 9(1). https://doi.org/10.21275/ART20203995

www.security.org

www.kaggle.com

pmc.ncbi.nlm.nih.gov

Safa, M. U., & Ganga, R. M. (2019). Credit Card Fraud Detection Using Machine Learning. International Journal of Research in Engineering, Science and Management, 2(11).

Saheed, Y. K., Hambali, M. A., Arowolo, M. O., & Olasupo, Y. A. (2020). Application of ga feature selection on Naive Bayes, random forest and SVM for credit card fraud detection. 2020 International Conference on Decision Aid Sciences and Application doi.org