Credit Card Fraud Detection Analysis - Intent Using SHAP and Predictive Models

Taan Gazi Safowan Islam
Dept of CSE
BRAC University
Dhaka , Bangladesh
taangazi.safowan.islam@g.bracu.ac.bd

Shaikh Faiyaz Shahriyer
Dept of CSE
BRAC University
Dhaka, Bangladesh
shaikh.faiyaz.shahriyer@g.bracu.ac.bd

***Abstract*—Credit card fraud is a major problem of the financial sector where large amounts of money are lost and consumer confidence is undermined. This project is meant to provide the best credit card fraud detection process using supervised machine learning techniques. The used dataset is very imbalanced, which is realistic since in real life fraudulent transactions are significantly rare compared to legitimate ones. To overcome this, the project applies several classifiers, such as: Logistic Regression, Support Vector Machine (SVM), Random Forest and XGBoost ensemble model together with the functionality of data preprocessing through StandardScaler which calculates the normalization of the features. Each model's performance is measured using precision, recall, F1-score and accuracy. Results show that ensemble techniques such as Random Forest and XGBoost perform well in identifying fraudulent transactions incurring minimal False Positive. This study views the importance of model selection and data preprocessing in the construction of reliable attacks detection systems.**

Index Terms–Credit card fraud, Machine learning, SHAP, Imbalance Data Handling, Ensemble Learning.

I. Introduction

As the cases of digital transactions continue to increase, the credit card has become a core element in modern commerce but at the same the digital transactions have increased instances of fraudulent activities. Credit card fraud does not only lead to significant financial losses of banking institutions, but also involves persons' trust and data integrity loss. When the traditional rule-based systems do not manage to change and-fit in with the changing nature of fraud, machine learning has been found out as a possible alternative for fraud detection in real-time and that is adaptive to change.

Machine learning algorithms can use past data to learn such complex patterns and anomalies that suggest that fraudulent behaviour is practiced. This project explains the integration of a few supervised learning algorithms including Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost, in order to develop a credit card fraud detection system. Since the datasets for fraud are highly imbalanced in nature, in which the normal transactions far outnumber the fraudulent transactions, this study also uses data preprocessing to include the operations of feature scaling, and evaluation approaches beyond accuracy, namely, precision, recall and F1-score.

The efficacy of methods of ensemble learning approaches to deal with such scenarios of imbalanced data set has been brought out by many recent studies. For example, Random Forest topped with oversampling techniques was proved to show good precision and recall for identifying the fraudulent transactions [1]. Indeed, also, the hybrid methods integrating such models as Random Forest, SVM, and Logistic Regression have been demonstrated to provide complementary strengths in terms of accuracy, AUC, and recall [2]. However, upon further comparative analysis by R. P. K. et al. [3], the use of the algorithms is supported since Random Forest was reported to be the most robust performer among numerous classifiers of fraud detection.

The goal of this project is to assess and compare these algorithms for a real-world dataset in order to choose the best model for the detection of credit card fraud without false positive rates and with practical implementability.

II. Literature Review

Credit card fraud detection has become an important area of machine learning (ML) application, especially with the modern dataset that is imbalanced in real life in that very few fraudulent transactions compose the overall dataset. Many studies have tried to look at and compare different ML techniques in an effort to overcome this challenge.

AM. Aburbeian and H. I. Ashqar proposed an Enhanced Random Forest Classifier that is tailored to deal with an imbalanced data problem for a fraud detection scenario [1]. Their method builds on the traditional Random Forest technique by the addition of weighted sampling and personalised thresholding techniques. The model was tested on real world imbalanced datasets and an improved capability to detect rare fraudulent transactions at the cost of

not degrading accuracy was maintained. This research illustrates the need for customizing traditional algorithms to reflect data skewness, an identifiable problem in the fraud detection environment.

G. Yang aimed at developing a machine learning pipeline for credit card fraud detection [2]. Yang compared a range of classifiers in this study; they were Decision Trees, Support Vector Machines, and Neural Networks. The results indicated that ensemble models, especially Random Forest and Gradient Boosting, produced greater robustness when applied to fraud prediction tasks. According to works reported in the research, hyperparameter tuning and cross-validation played a key role in enhancements made to the model performance. What's more, Yang emphasized the need for assessing models not only by accuracy, but by some other metrics such as precision, recall, F1-score, as the fraud data is highly imbalanced in nature.

A comparative analysis of several ML models conducted by R. P. K. et al. was used to establish the effectiveness of their use in detecting fraud [3]. This study compared algorithms, like Logistic Regression, k-Nearest Neighbors, Support Vector Machines, and Random Forests. From these, Random Forest and SVM had the best trade-off between decision rate and false positives. The authors also investigated effects of certain sampling techniques (SMOTE, undersampling) and discovered that synthetic oversampling greatly enhanced model's proficiency in detection of fraudulent transactions. This work supports the idea that the pre-processing and sampling methods are important in boosting the performance of the classification models upon imbalanced datasets.

Taken together, the results from these studies demonstrate that ensemble learning methods are effective in credit card fraud detection. They also teach everyone about the need to balance the data through specific points of sampling and stress upon the evaluation metrics that represent the real-life danger of such as false alarms and missed frauds. This review forms the basic criteria upon which appropriate models and preprocessing strategies can be chosen for the experimental phase of this project.

Alarfaj et al. [4] addressed the growing challenge of credit card fraud by evaluating both traditional machine learning and advanced deep learning techniques. Using the European card benchmark dataset, they compared models such as Decision Trees, Random Forests, SVM, and XGBoost with deep learning architectures, specifically Convolutional Neural Networks (CNNs). Their findings demonstrated that while traditional methods provided moderate accuracy, CNN-based models achieved significantly higher performance—an accuracy of 99.9%, precision of 93%, F1-score of 85.71%, and AUC of 98%. The study also employed data balancing strategies to mitigate false negatives, confirming that deep learning offers a more scalable and effective solution for fraud detection.

Xia [5] focused exclusively on the Support Vector Machine (SVM) model for fraud detection, emphasizing extensive hyperparameter tuning instead of algorithm comparison. Utilizing the Kaggle Credit Card Transactions Fraud Detection Dataset, the study incorporated preprocessing steps including feature transformation, scaling, and selection, ultimately reducing the dataset to 11 critical features. The optimized SVM model employed an RBF kernel and used GridSearchCV to fine-tune the regularization parameter (C) and kernel coefficient (γ), addressing issues of data imbalance and overfitting. While the model achieved a high AUC of 0.90, the F1-score remained relatively low at 0.260. The study underscored the importance of hyperparameter optimization in enhancing SVM performance on imbalanced datasets.

The issue of class imbalance remains a key concern in credit card fraud detection, where fraudulent transactions represent a small minority. Traditional techniques such as Bayesian Belief Networks, Artificial Neural Networks, and decision tree-based classifiers like C4.5 and CART have been widely applied, albeit with limited success. Meng et al. [6] conducted a comparative study using XGBoost applied to the original dataset, an undersampled dataset, and a dataset augmented with the SMOTE technique. Their results demonstrated that SMOTE and undersampling improved model recall and AUC, with SMOTE showing particularly strong performance in handling class imbalance. Furthermore, their analysis highlighted how XGBoost's capability to model feature interactions reduces the dependency on manual feature transformation like PCA, reinforcing its effectiveness in fraud detection scenarios.

III. Methodology

A. Dataset Description

The dataset that we used for our project consists of credit card transactions in the western United States. This Dataset was gathered between 21st January 2019 to 26th December 2020. It includes information about each transaction including customer details, the merchant and category of purchase, and whether or not the transaction was a fraud. The dataset is clean with no missing values in any of the columns and it presents no difficulties for its preprocessing and training of a model. Additionally, there are both numerical and categorical variables in the data, as well as geolocation-based features; therefore, it is quite convenient to conduct experimentation with both traditional machine learning algorithms and modern fraud detection techniques. The fact that there is the is_fraud label makes this a supervised binary classification problem where the key goal is to estimate the probability of a transaction being fraudulent by using the features associated with the transaction.

B. Data Preprocessing

To ensure data quality and compatibility with machine learning models,we made sure the dataset goes through a comprehensive preprocessing pipeline. Initially, we checked the dataset for duplicate entries using the .duplicated() method. We found a number of duplicate transactions and subsequently removed those with .drop_duplicates() to prevent bias and overfitting in the model.We also found that the columns trans_date_trans_time and dob were originally in string format. We converted it into datetime objects using pd.to_datetime() to enable time-based feature engineering. Furthermore, the target column is_fraud was found to contain string representations of numeric values (‘0’ and ‘1’). These were converted to integers to ensure proper classification processing. During inspection, we found that certain records in the is_fraud column contained incorrect values. We manually identified and corrected those to reflect valid binary labels (0 or 1). This step was critical in maintaining the integrity of the classification task. We also generated a correlation heatmap to visualize the linear relationships between numerical features. Highly correlated features (e.g., correlation > 0.9) might contain duplicate information. This redundancy can be removed to simplify the model and reduce overfitting. This gave a clear view to understand multicollinearity and understand how various features relate to each other and to the target variable is_fraud.

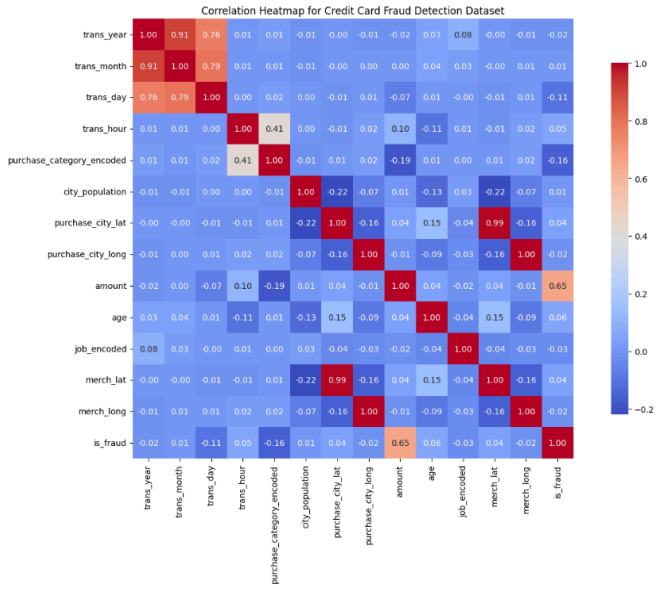

Fig. 1: Correlation Heatmap

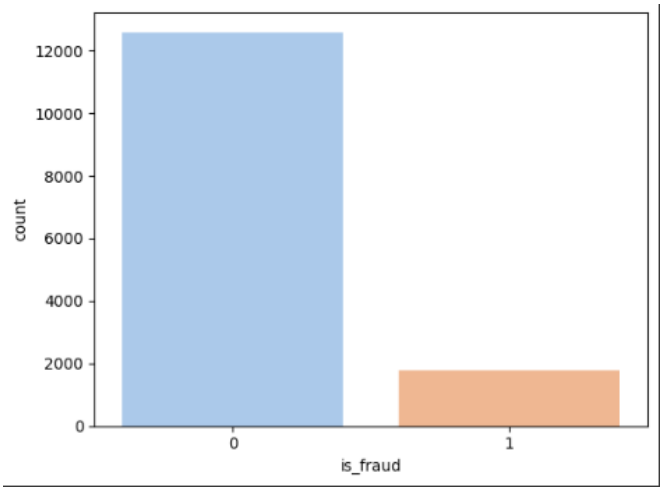

Fig 2: is_fraud Class Distribution

C. Learning phase:

The learning phase involves training and testing several machine learning models to identify the most effective one for credit card fraud detection. Initially, the cleaned dataset is divided into training and testing subsets using the train_test_split method, commonly in a 70:30 ratio. This division ensures that the model's performance is evaluated on unseen data, helping to measure how well it generalizes. A variety of classification algorithms are then chosen, considering their suitability for the task and how well they handle the dataset’s properties. Each model is trained using the training data, where it learns patterns and relationships between input features and the target labels. After training, the models are tested on the reserved testing set. Their performance is assessed using multiple metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Among these, the F1-score is prioritized for model comparison due to its balance between precision and recall—especially valuable for imbalanced datasets like those in fraud detection. It ensures the model is effective in identifying fraudulent transactions while minimizing false alarms. The study implemented the following classification algorithms:

1) Logistic Regression (LR): Logistic Regression is a fundamental classification algorithm that models the probability of a binary outcome based on one or more predictor variables. It uses the logistic function to ensure outputs lie between 0 and 1, making it ideal for binary classification tasks like fraud detection. In our project, Logistic Regression served as a baseline model due to its simplicity and interpretability.

2) Random Forest Classifier (RF): Random Forest is an ensemble method that creates multiple decision trees on training and uses the mode of the classes in classifying. It is especially good for dealing with high dimensional data, and

addressing the problems of overfitting. In our model, we employed the forest of 100 decision trees that improved predictive accuracy and resistance to the noise in the dataset.

3) Support Vector Machine (SVM): SVM is a powerful supervised learning algorithm that identifies the optimal hyperplane to separate classes in a high-dimensional space. It is particularly effective when dealing with non-linearly separable classes. In this project, SVM was used to learn complex boundaries between fraudulent and non-fraudulent transactions. However, it required careful tuning due to the large dataset size and the presence of class imbalance.

4) XGBoost Classifier (XGB): XGBoost is an advanced boosting algorithm which is designed for speed and performance. It develops models sequentially and prioritizes the correction of mistakes of the previous models. We applied XGBoost because it is able to deal with class imbalance and has a better performance on structured data. Some hyperparameters, such as the learning rate and the number of estimators were tuned to maximize performance.

The performance of each model was measured on a test dataset and the results highlighting accuracy, precision, recall, F1-score and AUC are discussed further in the Results and Discussion portion of this report.

D. SHAP(SHapley Additive exPlanations):

SHAP (SHapley Additive exPlanations) is a powerful framework for interpreting predictions made by machine learning models. It is based on cooperative game theory and uses the concept of Shapley values to fairly attribute the contribution of each feature to a particular prediction. SHAP allows us to understand not only which features are most important globally (i.e., across the whole dataset), but also how individual features impact a specific prediction. This is especially valuable in sensitive domains such as fraud detection, where understanding why a model flagged a transaction as fraudulent is as important as the prediction itself. In our project, we used SHAP to interpret the results of the best-performing model—XGBoost. The use of SHAP enhanced the transparency of our model by providing visual explanations of how input features such as transaction amount or merchant information influenced fraud predictions. This interpretability is critical for gaining stakeholder trust and for further refining the model based on actionable insights.

IV. Results

Our project aimed to implement a variety of machine learning models to accurately detect fraudulent financial transactions, along with utilizing explainable AI (XAI) techniques such as SHAP for comprehensive model interpretability.

A. Model Performance Comparison

Four models were evaluated during experimentation: Random Forest, Linear Regression (adapted for classification), Support Vector Machine (SVM) with hyperparameter optimization, and XGBoost. Among these, XGBoost delivered the most impressive results, achieving an overall accuracy of 98% and an F1-score of 92%, indicating strong performance in identifying both fraudulent and non-fraudulent cases.

Random Forest also performed well, with 97% accuracy and an F1-score of 85%, making it a strong contender. However, Linear Regression, despite being scaled and thresholded for classification, struggled with the minority class, achieving the lowest recall and F1-score, which makes it less suitable for imbalanced datasets like fraud detection. SVM, while moderately successful with tuned parameters, still lagged behind in identifying fraud cases compared to tree-based models.

In summary, XGBoost was chosen as the best-performing algorithm, not only because of its high overall accuracy but also due to its balanced performance across precision, recall, and ROC-AUC. The comparative results of all models are shown in Table I, with further visual insights presented in Fig. 3 and Fig. 4, highlighting the significant difference in fraud detection capabilities among the models.

Table I

Model Evaluation Metrics

Model	Accuracy	Precision	Recall	F1-Score	Roc-Auc
RF	97%	0.94	0.78	0.85	0.9833
LR	94%	0.91	0.55	0.68	0.8875
SVM (Tuned)	93%	0.66	0.83	0.74	0.9414
XGBoost	98%	0.94	0.91	0.92	0.9929

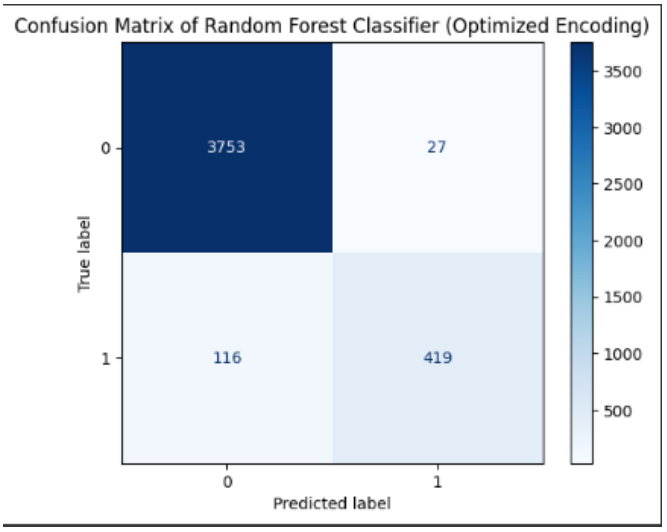

Fig 3: Confusion Metrix - Random Forest

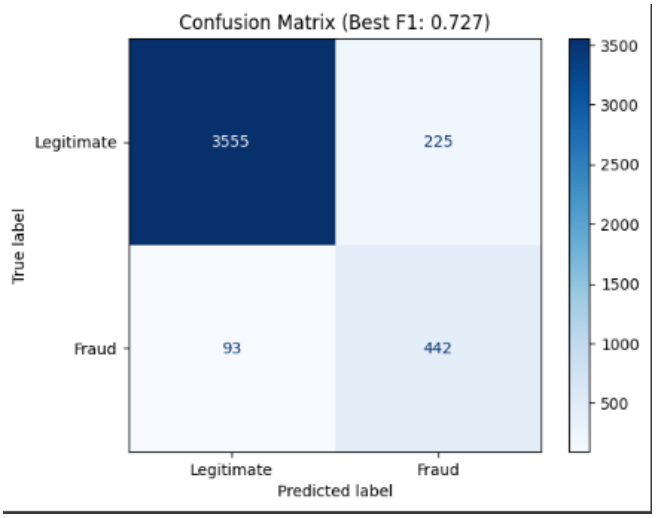

Fig 4: Confusion Metrix - SVM

B. SHAP Analysis Outcome

We applied SHAP to the XGBoost model, which emerged as the best-performing algorithm. The SHAP summary plot (Fig. 5) revealed that transaction amount was the most influential feature in determining the likelihood of fraud. This was followed by encoded merchant-related variables, suggesting that certain merchants were more associated with fraudulent behavior. Features like purchase city and purchase state had moderate to low influence on the model's predictions, indicating that geographical data played a lesser role in fraud detection compared to financial and behavioral variables. The SHAP violin (beeswarm) plot (Fig. 6) provided further insights. It showed that higher transaction amounts positively contributed to fraud predictions — indicating a high probability of fraud — while lower amounts contributed negatively. This suggests that the model correctly learned that fraud is more likely to occur in high-value transactions. By utilizing SHAP, we not only improved model transparency but also gained valuable domain insights. For example, certain merchant patterns or location-frequency anomalies might be indicative of fraud rings or operational vulnerabilities.

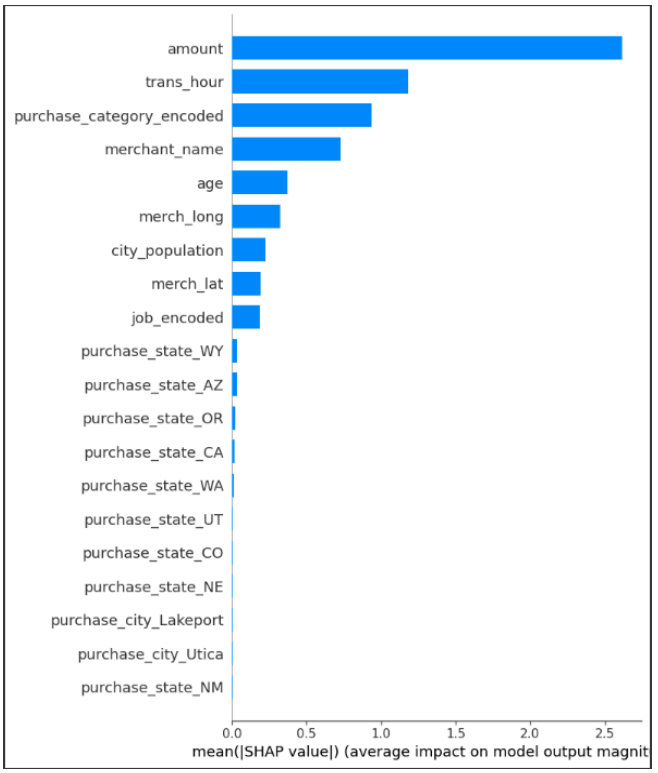

Fig 5: SHAP Feature Importance

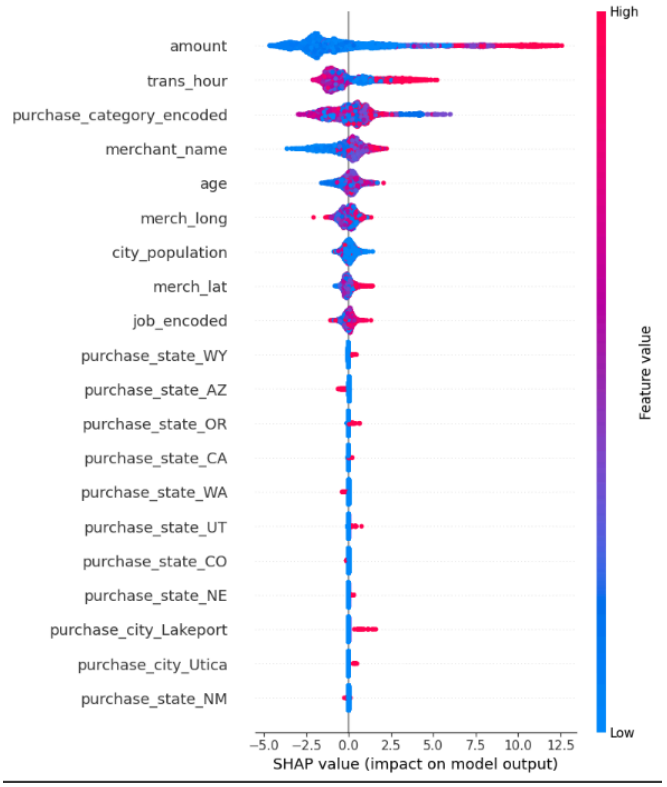

Fig 6: SHAP Summary Plot

V. Conclusion

In this project, we developed and compared multiple machine learning models to detect fraudulent transactions using a real-world dataset. The preprocessing pipeline involved encoding nominal and high-cardinality categorical

features using One-Hot Encoding and Target Encoding respectively. We paid special attention to data imbalance, which is a common challenge in fraud detection, by applying stratified sampling and using class-weighted models. Among all models tested, XGBoost emerged as the best-performing classifier with an impressive ROC-AUC of 0.9929, indicating excellent ability to discriminate between fraudulent and legitimate transactions. Tree-based methods like XGBoost and Random Forest demonstrated clear advantages in handling non-linear relationships and interactions between features. We further enhanced our analysis by applying SHAP to interpret the predictions of the XGBoost model. This interpretability layer allowed us to confirm the model's decision logic and provided actionable insights into which features most influenced fraud predictions. Our project highlights the importance of combining predictive accuracy with interpretability in sensitive applications like fraud detection. A high-performing model is only truly valuable when it is explainable and trustworthy—qualities that we achieved through both model tuning and SHAP-based interpretation. This framework can be used not only to detect fraud but also to inform prevention strategies, optimize monitoring systems, and guide investigations. As future work, the model can be deployed in real-time fraud detection systems, and enhanced further using ensemble methods, temporal features, or deep learning architectures.

References

[1] A. M. Aburbeian and H. I. Ashqar, "Credit Card Fraud Detection Using Enhanced Random Forest Classifier for Imbalanced Data," *arXiv preprint arXiv:2303.06514*, 2023. doi: 10.48550/arXiv.2303.06514

[2] G. Yang, "Credit Card Fraud Detection Based on Machine Learning Prediction," in *Proc. 2024 2nd Int. Conf. Image, Algorithms and Artificial Intelligence (ICIAAI)*, Atlantis Press, 2024, pp. 35–45. doi: 10.2991/978-94-6463-540-9_5

[3] R. P. K., R. Mathew, A. Walawalkar, P. Patil, U. Shirode, and A. Gaadhe, "A Comparative Analysis of Machine Learning Techniques for Detecting Credit Card Fraud," *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 3, pp. 146–153, 2024. [Online]. Available: https://ijisae.org/index.php/IJISAE/article/view/5232

[4] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022, doi: 10.1109/ACCESS.2022.3166891.

[5] J. Xia, "Credit Card Fraud Detection Based on Support Vector Machine," *Highlights in Science, Engineering and Technology*, vol. 23, pp. 93–97, 2022, doi: 10.54097/hset.v23i.3202.

[6] C. Meng, L. Zhou, and B. Liu, "A Case Study in Credit Fraud Detection With SMOTE and XGBoost," *Journal of Physics: Conference Series*, vol. 1601, no. 5, p. 052016, 2020, doi: 10.1088/1742-6596/1601/5/052016.

Word Distribution:

1. Shaikh Faiyaz Shahriyer – 24141235

a. Dataset Pre Processing Combined
b. Feature Encoding
c. Column drop based on Heatmap Generation
d. Implementing SVM (Tuned)
e. Implementing XG Boost
f. SHAP Framework Implementation and Graph Generation

2. Taan Gazi Safowan Islam – 21201384

a. Dataset Upload
b. Dataset Explanation and Analysis
c. Dataset Preprocessing Combined
d. Dataset Analysis Graphs
e. Impleenting Random Forest Classifier
f. Implementing Linear Regression classifier