Logistic Regression-Based Real-Time Credit Card Fraud Detection Midterm Report

1st Anant Sahoo
EECS
University of Tennessee, Knoxville
Knoxville, TN, USA
asahoo@vols.utk.edu

2nd Deep Patel
EECS
University of Tennessee, Knoxville
Knoxville, TN, USA
dpate125@vols.utk.edu

3rd Sulaiman Mohyuddin
EECS
University of Tennessee, Knoxville
Knoxville, TN, USA
smohyud1@vols.utk.edu

***Abstract*—Credit card fraud represents a substantial financial threat in today's increasingly digital economy. This study introduces a logistic regression (LR) model for real-time detection of potentially fraudulent transactions, employing a publicly available dataset from Kaggle. Preprocessing steps include addressing missing values and scaling numerical features. For model validation, we employ k-fold cross-validation to split the data into training and testing sets. Achieving an accuracy of 92%, the model is evaluated through confusion matrices and accuracy scores. Logistic regression was selected over more complex methods, such as Support Vector Machines (SVMs) and Neural Networks (NNs), due to its computational efficiency and suitability for real-time applications. Future work aims to enhance performance through optimized feature engineering, dataset stratification, and improved scalability. The platform's real-time notifications are designed to help mitigate the impact of fraudulent activities on individuals and organizations alike.**

***Index Terms*—Credit Card Fraud Detection, Logistic Regression, Model Evaluation, Real-Time Systems, Data Preprocessing**

I. Introduction

The use of credit cards for financial transactions has become commonplace in the United States. This ever-growing digital shift, however, gives attackers plenty of opportunities to steal information and commit fraud through a variety of sources. Recent data shows that 60% of Americans have experienced at least one unauthorized charge, which equates to around 128 million people. The magnitude of these charges is not minuscule either because in the last two years, the median has increased by 26%, from $79 to $100 [1]. So what can be done to mitigate these damages? While directly securing these systems is a crucial step in the process, detecting fraud as soon as possible will allow users or companies to take immediate, preventative action.

A. Project Goals

We aim to provide an accessible, inclusive platform powered by a Logistic Regression (LR) model that notifies users of potential fraud in real-time. The two key objectives of our project are as follows:

1) *High Accuracy*: A real-time detection platform would provide little utility to users with sub-optimal accuracy. False positives create high uncertainty, and false negatives greatly increase the risks of further harm. Both scenarios lead to headaches and financial loss.
2) *Real-time Detection*: Time is money. Our platform intends to communicate through email and mobile notifications to reach a wide audience. To ensure this, we require low-latency servers and high computational power for our LR classifier.

By achieving these goals, the project will provide a practical tool to help mitigate the impact of credit card fraud on individuals and businesses.

II. Dataset and Data Preprocessing

The dataset used in this project, sourced from Kaggle, contains information on over 200,000 credit card transactions from the Western United States, with a small fraction labeled as fraudulent. The data includes anonymized transaction features, encompassing customer details, merchant and purchase categories, and an indicator of whether each transaction was fraudulent. Preprocessing steps included:

- **Handling Missing Values:** We identified and removed any null values in the dataset to ensure consistency and integrity during model training.
- **Label Cleaning:** We filtered out incorrectly labeled samples and converted categorical target labels (0 for ”not fraud” and 1 for ”fraud”) to integers for compatibility with machine learning algorithms.
- **Location Consolidation:** To reduce dimensionality, we combined city and state fields into a single location feature, enhancing the simplicity of the model while preserving location information.
- **Date Conversion:** We transformed both transaction date and date of birth fields from date formats to numerical timestamps, allowing the model to interpret them effectively as temporal data.
- **One-Hot Encoding and Feature Extraction:** We applied one-hot encoding to categorical features (category, merchant, job, and location) to convert them into numerical representations. After encoding, we removed the last column of each encoded feature to avoid redundancy. Finally, we rearranged the is_fraud column as the target variable at the end of the DataFrame.

A. *Exploratory Data Analysis*

We perform some baseline analysis of the feature data, focusing on the “category” feature. Out of the 14 unique categories, some common ones included grocery, gas, and shopping. However, the distribution of these categories was evenly distributed for the most part, as demonstrated in Figure 1. Some more interesting patterns arise when examining the relationship between categories, purchase amounts, and fraudulent proportions. Firstly, fraudulent purchases are strikingly more expensive than non-fraudulent ones. For the entire dataset, the median fraudulent purchase was $358.66, compared to a measly $46.14. When we filtered out by categories, certain ones revealed a higher proportion of fraudulent purchases than others. Shopping, entertainment, and grocery categories showed some of the highest rates of fraudulence, while travel, personal care, and food dining showed some of the lowest. Figure 2 displays transaction amount versus category for both fraudulent and non-fraudulent purchases.

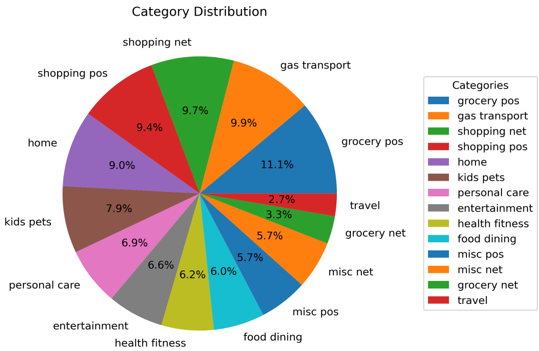

Fig. 1: Category Distribution of Transactions

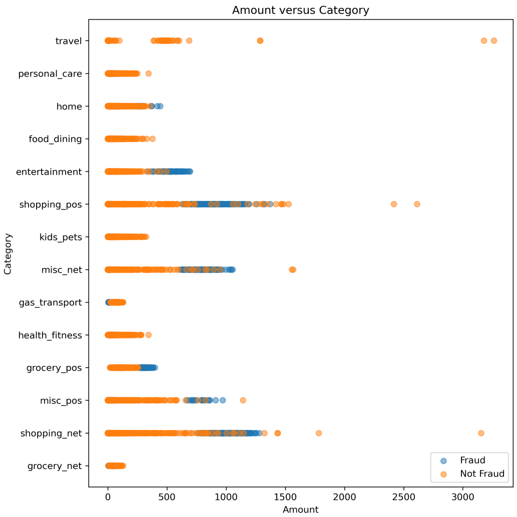

Fig. 2: Amount versus Category Scatter Plot

III. Baseline Model and Related Work

Researchers in machine learning and artificial intelligence have studied and implemented various models and methodologies that aim to solve the fraud detection problem. One such model that has been explored is the support vector machine (SVM). These models tend to utilize features that represent transactions and are used in the clustering process. The SVM then isolates these features based on similarities, and each of these isolations (clusters) is used for the final classifier [2]. Neural network-based models have also shared a spotlight amongst fraud detection classifiers. Artificial Neural Networks (ANNs) consist of an input layer, hidden layers, and an output layer. They are initially trained with the normal behavior of a cardholder, and then the suspicious transactions are backpropagated through the NN where a final prediction is completed [3]. NNs, however, require high processing time, especially as the complexity of the NN scales. While it is certainly worthwhile to meticulously weigh the costs and benefits of these various solutions, we choose to select an LR model as our classifier for its good baseline performance and high computational efficiency.

A. *Model Implementation*

Alenzi et al. [2] is the foundation for our baseline LR model. This section outlines the steps after the data is cleaned. To start, the tools we use are the standard Python scientific computing and machine learning libraries which include NumPy, pandas, sci-kit learn, and scipy. Our final model is based on the cross-validation technique, where we use k=10 folds. For each fold, the process works as follows:

- Training and Testing: We have $N = 14,444$ total samples, and the number of samples per fold can be calculated as $\frac{N}{10}$. For the i-th fold, the training set includes all samples outside the range $[i \times \text{num\_samples\_per\_split} : (i+1) \times \text{num\_samples\_per\_split}]$, while the testing set comprises this range. This results in a standard 9:1 ratio of training to testing sample size for both the features (X) and the labels (y).
- Once the split is complete, we initialize an LR model using the training data and store the X-test, X-train, y-test, y-train, and the newly fitted model in an array for later use.

After the 10 models and their corresponding data splits have been generated, we unpack each tuple containing the testing and training data and the pre-initialized model. The model is then trained on the training data and generates its prediction on the testing data.

IV. Evaluation

To find the total accuracy of the model, we calculate the arithmetic mean of the accuracies of each fold.

$$\text{ACCF}_C = \frac{1}{10}\sum_{k=1}^{10}\text{Acc}_k$$

The model achieved an accuracy of 92%, which aligns with industry benchmarks for real-time fraud detection. Other key performance metrics include:

- **Confusion Matrix:** We built a confusion matrix for each fold and averaged those results into one final matrix, as shown in Figure 3. Here, the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates can be attained. The FP rate is especially critical to our model because it represents the proportion of samples predicted to be fraudulent when in reality they were non-fraudulent. A system that alerts users of frequent FPs is unreliable and causes unnecessary worry. Our model was able to keep this rate at around 1%.

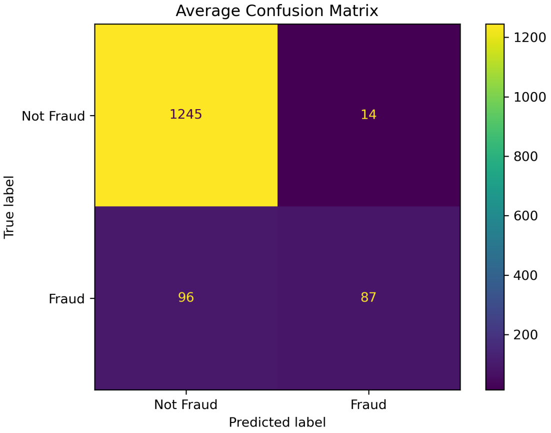

Fig. 3: Average Confusion Matrix

$$\text{False Positive Rate} = \frac{\text{FP}}{\text{FP} + \text{TN}} = \frac{14}{14 + 1245} \approx 0.01$$

V. Proposed Improvements

Despite the promising results of our baseline Logistic Regression (LR) model, several improvements can be made to enhance both accuracy and model performance:

- **Scaling the Features:** Currently, the model cannot process the transaction number (*trans_num*) column due to scikit-learn's limitations. Future iterations will ensure all columns are scaled appropriately, allowing full feature utilization and improving overall model accuracy.
- **Feature Selection:** We will apply advanced feature selection techniques to identify the most relevant features contributing to the model's performance. We can do this by applying Sequential Backward Selection (SBS). By reducing noise in the data, we can further improve accuracy and model interpretability.
- **Feature Extraction:** Feature extraction methods, such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), will be explored to reduce dimensionality and capture more meaningful relationships within the data.
- **Stratification:** The class imbalance present in the dataset requires further attention. Implementing stratified sampling will ensure that both the training and test sets maintain similar fraud-to-non-fraud ratios, improving the robustness and generalizability of the model across various datasets.

VI. Conclusion

The Logistic Regression model presented in this report demonstrates strong potential for real-time credit card fraud detection, achieving an accuracy of 92% with competitive computational efficiency. Using preprocessing techniques and k-fold cross-validation, we validated the model's effectiveness. However, improvements such as better feature scaling, selection, and stratification are necessary to address limitations in handling imbalanced datasets and underutilized features. Future work will also explore feature extraction methods to optimize performance further. The proposed enhancements will ensure the model's scalability and suitability for deployment in real-world financial systems to mitigate fraud risks.

VII. Distribution of Work

- **Anant Sahoo:** Created and Prepared the basic layout of the Latex, worked on the abstract section, data preprocessing section, evaluation section with Sulaiman, improvements section, conclusion section, report editing, and pair programming with Deep.
- **Deep Patel:** Developed data analysis code with Anant and Sulaiman by side, implemented the LR model, conducted performance assessment, created figures, and worked on the model implementation section with Sulaiman.
- **Sulaiman Mohyuddin:** Worked on the introduction section, data analysis section, baseline model, model implementation with Deep, evaluation with Anant, report editing, and pair programmed with Deep.

References

[1] Security.org, "Credit Card Fraud Statistics and Reports", 2023. [Online]. Available: https://www.security.org/digital-safety/credit-card-fraud-report/.

[2] H. Z. Alenzi and N. O, "Fraud Detection in Credit Cards using Logistic Regression," International Journal of Advanced Computer Science and Applications, vol. 11, no. 12, 2020, doi: https://doi.org/10.14569/ijacsa.2020.0111265.

[3] C. Liu and D. P. Robinson, "A Comparison of Machine Learning Models for Fraud Detection," *IEEE Trans. Neural Networks*, vol. 29, no. 4, pp. 987–998, 2019.

[4] Kaggle, "Credit Card Fraud Detection Dataset," [Online]. Available: https://www.kaggle.com/datasets/neharoychoudhury/credit-card-fraud-data/data.