Credit Card Fraud Detection

Academic Year:2025-2026

Course: Machine Learning (4S010673)

Master in Artificial Intelligence

Author Name: Muneeb Ahmed Khan

VR Number: 545124

Author Name: Gautam Parmar

VR Number: 545120

Table of Contents

1. Motivation & Rationale

Our project is going to determine whether or not the transaction is legitimate or fraudulent by using the Logistic, Random Forest, and Support Vector Machine (SVM) models. We have developed a way to address the issue of an imbalanced dataset since fraudulent transactions are quite uncommon. The model is trained using legitimate transactions the majority of the time, which does not enable the model to accurately detect fraudulent transactions. Therefore, the issue of an imbalanced dataset can be a problem for the development of an effective detection system. The imbalance is an important consideration for a bank because if the model is trained using an unbalanced dataset, the majority of transactions will be labelled as legitimate. As a result, the model will begin to misinterpret fraudulent transactions, which would prevent the bank from accurately tracking transactions and may lead to a loss of confidence from its clients. We have found that lots of the models in the current solutions are using historical data and don't know how to tackle imbalanced data. Despite their accuracy, they will still fail due to the old method.

2. State of Art

Transaction fraud is still a big problem for banks and other financial organizations since it doesn't happen very often, the ways individuals do it are continuously evolving, and they need to be able to discover it quickly and correctly. Machine learning (ML) is becoming a major way to look at large volumes of transactional data and adapt to new types of fraud. However, a lot of the literature changes the natural class distribution by resampling, which might make stated performance look better than it really is and make it less effective in the real-world scenario [1]. As the digital economy increases, it becomes tougher to catch fraud. This means that methods for finding things need to be more advanced, adaptable, and easy to use. Finding things by hand in the old-fashioned manner doesn't work. The banking industry is slowly starting to embrace machine learning (ML) and understandable artificial intelligence (XAI) technologies [2] to discover fraud more readily and stay strong. The European Central Bank and the European Banking Authority indicate that more than 70% of fraudulent payment transactions in the EU in the previous several years happened online, through activities like phishing, account takeovers, and card-not-present (CNP) fraud [3]. These days, using supervised classification algorithms trained on data from prior transactions is the best approach to discover fraud. Logistic regression, Support Vector Machines (SVM), and Random Forest models are the main parts of realistic fraud-detection pipelines.

Firstly, we adopted the Random Forest model as our primary method for fraud detection. Random Forest is a supervised machine learning approach that employs an ensemble of decision tree models for classification and prediction tasks. Each decision tree is a poor learner due to its limited predictive capacity. It is based on ensemble learning, which uses a lot of decision tree classifiers to solve a problem and make the model more accurate. As a result, the random forest uses a bagging method to make a group of decision trees [4]. In the context of classification, the random forest classifier m is derived using a majority vote among K classification trees using input x, namely,

$$m(x:\theta_1,\ldots,\theta_k) = \begin{cases} 1 & if \\ 0 & otherwise \end{cases} \frac{k}{1}\sum_{j=1}^{k} m(x;\theta_j) > \frac{1}{2} \quad [7]$$

Nonetheless, Random Forests exhibit inherent limitations characteristic of supervised learning in fraud detection: they rely significantly on balanced training data, and they deteriorate as fraud patterns evolve, as well as encounter difficulties in identifying exceedingly unusual fraud categories that infrequently arise in past transactions.

On the other hand, a model which is SVM has also been introduced in our project for fraudulent detection, which treats detection as a binary classification problem. Support Vector Machines (SVM) are utilized in classification and pattern recognition applications, including facial recognition, bioinformatics, and text categorization. The Risk Minimization hypothesis is formulated and endorsed by Support Vector Machines (SVM). Support Vector Machines (SVM) appear to be a widely utilized machine learning technique, with several effective implementations, including XGBoost. Despite the incorporation of several software optimizations in these implementations, speed and scalability remain inadequate when the function dimension is elevated and the data amount is substantial. A primary issue is that they must examine all instances of data for each characteristic to evaluate all the data gathered from all possible splitting positions, which is time-consuming. Support Vector Machine (SVM) was introduced to resolve this problem [5]. However, this increased flexibility engenders new vulnerabilities: Support Vector Machines (SVMs) exhibit suboptimal scalability when handling millions of transactions typical in actual financial networks, and their approach to decision-making is substantially less interpretable compared to logistic regression. As transaction volumes grow and demands for transparency rise, SVMs alone cannot meet the dual objectives of effectiveness and understanding.

These limitations inherently result in the utilization of ensemble-based techniques, notably Logistic Regression, which epitomize the current practical state-of-the-art for several fraud-detection frameworks. A logistic regression-based classifier was proposed to construct a classifier that helps stop credit card fraud detection [6]. Logistic Regression seems to be the most valuable model for fraud detection, as This regression model features a categorical response variable Y. Logistic regression enables the estimation of the likelihood of a categorical answer based on one or more predictor variables, x. It enables one to

assert that the existence of a predictor modifies the probability of a certain event by a defined percentage. Logistic regression mathematically estimates a multiple linear regression function expressed as:

$$Y_i = \beta_0 + \beta_1 x_i,1 + \beta 2\ x_i,2 + ... + \beta_p\ x_i,\ p \qquad [7]$$

However, it is evident that fraud does not always follow a linear path. When the correlations between variables get intricate or people start to act dishonestly over time, logistic regression has problems.
The three most effective models share a common issue: they exhibit an abundance of imbalanced classes and fail to prioritize concepts. Usually, algorithms guess "legitimate" until they address the problem because only a small part of the historical data is made up of false transactions. Furthermore, deceptive methods evolve frequently; thus, any fixed model, whether a linear, margin-based, or ensemble model, would always be less accurate, even with little changes. These issues illustrate how crucial it is to make both the data handling and the model building better.

The creation of this project solely reflects these state-of-the-art analyses by finding the SVM, Random Forest and Logistic Regression algorithms under a real-time fraud detection scenario. It aims at resolving the flaw which was found above with the following methods, which are resampling for imbalance, preprocessing of features and hyper parameter tuning for better strength and generalization. Through comparison of these models on the similar dataset and limitation, the project found which state-of-the-art methods remain useful, which break down beneath real-world imbalance, and for reliable fraud detection, what changes are required.

3. Objectives

1. To determine if a transaction is legitimate or fraudulent using an imbalanced dataset.
2. Make an observation on the difference after the class distribution using the under sampling method.
3. A comparison may be made between the accuracy of three distinct categorization models.

4. Methodology

The methodology section contains following steps:

4.1 Dataset Description:

The Dataset has been imported from publically available Kaggle. The format of the dataset is a .CSV (Comma Separated Values) file. Prepare the data by removing duplicates and verify that the dataset contains no missing values through card_data.info(). Data size contains approximately 284,807 rows and 31 columns. The columns highlight the features, including time in seconds and transaction amount, while

the v1-v28 columns contain details about account holders, all of which carry numeric values. The Target variable is 0 which represent Legit Transaction and 1 which represent Fraudulent Transaction and both values are present under Class label. Only 492 fraudulent transactions are present in the dataset, which shows that the dataset is highly imbalanced.

4.2 Data Preparation

In the selected dataset, all features are present as numerical values, so there is no encoding required for Data preprocessing. Moreover, the dataset did not contain any null values, so no imputation step has been choosing, whereas features like amount and time need a scaling process. Additionally, in data preparation, we have examined class distributions and correlation patterns. As data is already compressed through PCA, the interpretation of features is limited, and the dataset was split into Feature (X) and Target Label (Y). To tackle the imbalanced data set, we use the undersample process for equal class distribution and split the data into the ratio of 80% and 20%.

1. Model Parameters

The following parameters were taken for model training and manage their process throughout training and assessment.

a) Logistic Regression Setup

We used logistic regression as one of the benchmarks for fraud detection, initializing the algorithm with Scikit-Learn parameters to assess its initial behavior before heavy tuning. Moreover, we did not apply an encoding step because all features are numerical. To avoid overfitting, the dataset was split into an 80% training ratio and a 20% test ratio, and with 0.94 accuracy in both phases, consistency was observed. Linear Regression allows to determine the performance are actually enhancing or just including needless complexity.

b) Random Forest Implementation

The random forest algorithm was used as a nonlinear ensemble model to increase performance in comparison to simpler classifiers. This model has been executed with 200 trees, which help to steady predictions and minimize overfitting, while we have set class weight on balanced, by which the model will automatically modify tree splits to handle imbalanced classes earlier. To train and test the Random Forest Model, we have taken the same ratio as we take in Logistic regression. The selection of random forest is because of Fraud pattern which are mostly irregular and difficulty to process under any other classifier.

c) Support Vector Machine Setup

The support vector machine model generally helps deal with difficult classifiers, which allow the model to separate legitimate and fraudulent transactions by using nonlinear boundaries. We have set the SVM kernel on RBF which allows the algorithm to observe nonlinear patterns that are present in

the dataset, whereas by adjusting the probability on True, we can generate ROC curves and evaluations which are based on probabilities while to maintain consistency and for comparison purposes, the model was split within the same ratio as of Logistic Regression and Random Forest. Additionally, we also found that SVM has good control over small and imbalanced datasets and gives a different learning view as compared to other linear and tree models.

4.3 Evaluation Approach

To assess the performance of the prediction model, we utilized multiple performance indicators, such as accuracy, precision, and recall. The results of the categorization process were set and presented in a confusion matrix, as depicted in Table 01. This matrix shows the distribution of accurate and inaccurate predictions for each category, with the counts highlighting the percentage of instances falling into each cell. The confusion matrix uses the labels (P, N) to represent the positive and negative testing data, respectively. The labels (Y, N) indicate the classifier's predictions for the positive and negative classes, as explained in reference.

	Actual Positive (P)	**Actual Negative (N)**
Predicted Positive (Y)	TP (true positives)	FP (false positives)
Predicted Negative (N)	FN (false negatives)	TN (true negatives)

Table 2. Confusion matrix

The confusion matrix helps to differentiate between the number of correct predictions for positive outcomes, called as “True Positive: and the number of correct predictions for negative sample, referred as “True Negative”. Conversely, the number of incorrect predictions for positive outcomes is known as “False Positive”, whereas, the number of incorrect predictions for negative sample is referred as “False Negative”. To evaluate the efficiency of the prediction algorithm using the dataset, we use a set of evaluation metrics as mentioned below.

Accuracy:

$$\frac{TP + TN}{TP + FP + FN + TN}$$

Precision:

$$\frac{TP}{TP + FP}$$

Recall:

$$\frac{TP}{TP + FN}$$

F-measure:

$$\frac{(1 + \beta^2) * Recall * Precision}{\beta^2 * Recall + Precision}$$

The factor β in the equation behaves as a coefficient that is used for fine-tuning the trade-off between recall and precision. The F-measure is frequently provided a value of 1, which confirms that both recall and precision are providing uniform importance in its calculation. A high-level F-measure value shows that the target class carries the performance of the learning method well because it means that both precision and recall values are high. Select simply, a high F-measure shows that the model is efficiently identifying an important fraction of true positive cases while also carrying the value of false positive predictions low.

4.4 Tools and Software

We have used Python as the main language for our project, whereas Pandas and Numpy libraries are important for cleaning, manipulation, and data loading. For Machine Learning Framework we utilized Scikit-Learn, which also performs train and test splitting and evaluates metrics. For visualization through confusion metrics, ROC curves and comparison of model performance, we use Matplotlib and Seaborn. The overall experiment was carried out on google colab.

5. Experimental Result

This section will describe about the performance of Logistic Regression, SVM and Random Forest has been assessed for the detection of Fraud or Legit transaction. All trials were led on the balanced dataset which was created by using undersampling process with the ratio of 80% and 20% train-test split. Performance of algorithm was conduct through multiple metrics, that include recall, precision, F1-score, confusion matrix and AUC, as alone accuracy is inadequate for imbalanced classification issue. Furthermore, we have discussed our observation in detail below.

5.1 Overall Performance Analysis

We have evaluated the performance of all models using multiple metrics, as shown in Table 01. Random Forest and Logistic Regression perform better throughout most metrics, whereas SVM executes unwell as compared to the other two models. Recall and F1-score are more essential metrics than accuracy for fraud detection, so for Logistic both F1 and recall give balanced value, while Random Forest has higher precision but considerably lower recall, but SVM was not found suitable for this dataset.

Models	**Accuracy**	**Precision**	**F1- Score**	**Recall**	**AUC**
Logistic Regression	0.94	0.97	0.94	0.91	0.97
Random Forest	0.93	1.00	0.92	0.86	0.97
SVM	0.51	0.51	0.50	0.48	0.51

Table. 02. Model Performance Comparison

1.2 Classification Behavior

Using the confusion matrix, we analyzed the classification performance of the algorithms for detecting legitimate and fraudulent transactions. As Figure 01 indicates, both classes have been correctly classified by the Logistic Regression algorithm by showing a balanced performance with fewer misclassifications, whereas Figure 02 indicates that Random Forest achieves solid classification for legitimate transactions but misses a high level of fraudulent cases, which highlights the exchange between recall and precision. However, Figure 03 shows that the SVM algorithm creates a significant number of misclassifications for both classes by specifying weak separation between legit and fraudulent transactions. These results emphasize that random forests and logistic regression perform very well for fraud detection, whereas SVM is not appropriate for this detection.

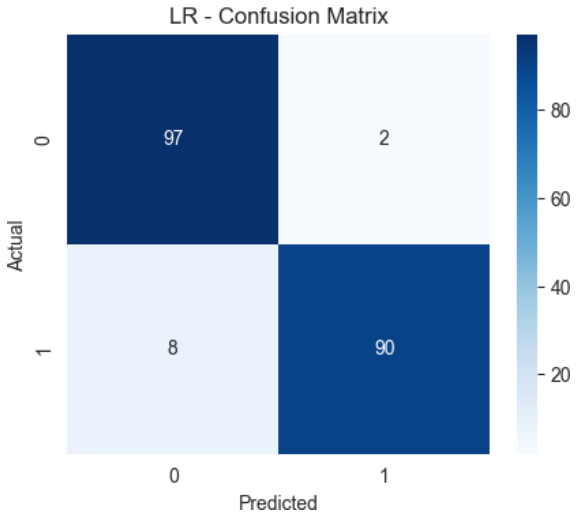

Fig. 01. Confusion Matrix for Logistic Regression

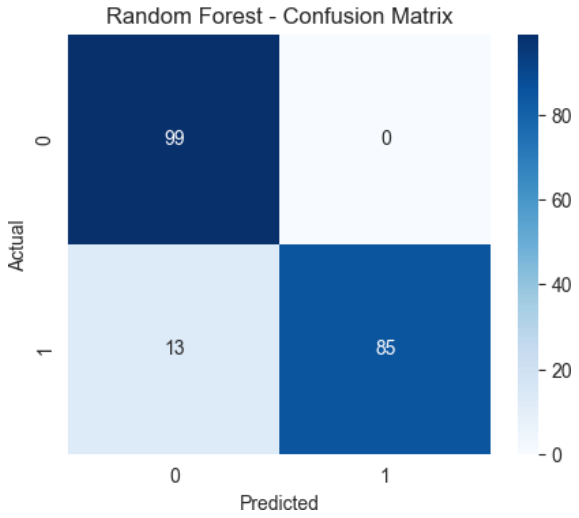

Fig. 02. Confusion Matrix for Random Forest

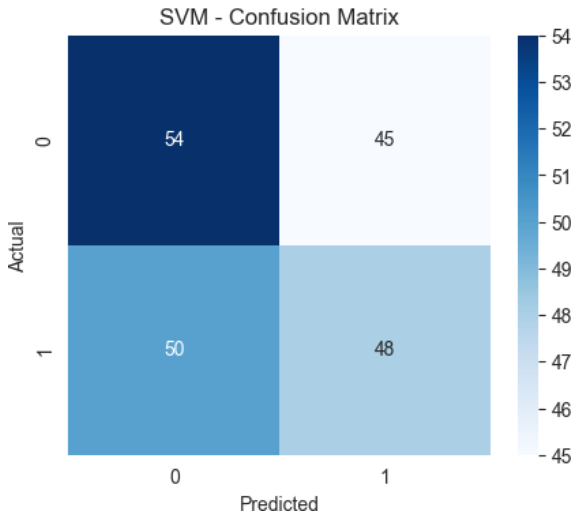

Fig. 03. Confusion Matrix for Support Vector Machine

1.3 ROC and Distinctive Capability

To identify the distinctive capabilities of all three models, we have used ROC curves, which help distinguish fraudulent and legitimate transactions. Logistic regression describes solid separation of class, as the ROC curve remains near to the top-left corner, which highlights consistent discrimination performance, as indicated in Figure 04. Moreover, the same pattern has been perceived for the Random Forest algorithm that also gains high-level discriminative ability through different point values, as highlighted in Figure 05. In comparison, the ROC curve of the SVM algorithm rests near to the diagonal position line, which suggests performance near to guessing randomly and weak class separability, as shown in Figure 06. These analyses endorse that Fraud and non-fraud transactions are successfully separated by Linear Regression and Random Forest while SVM does not meet the classification requirement.

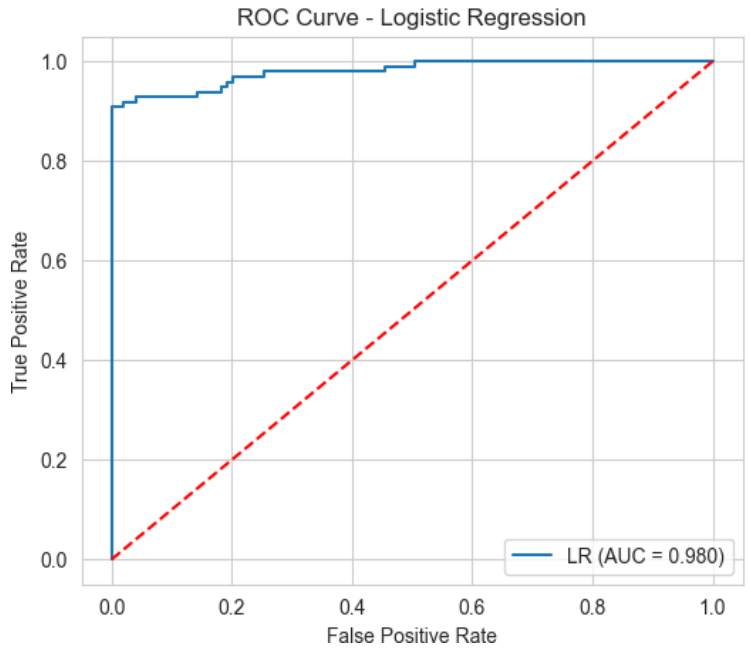

Fig. 04. ROC Curve for Logistic Regression

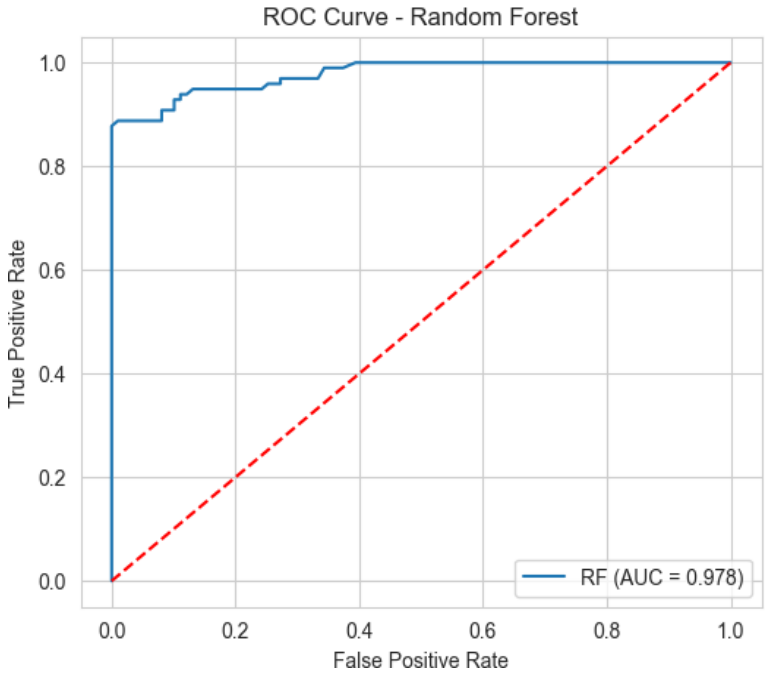

Fig. 05. ROC Curve for Random Forest

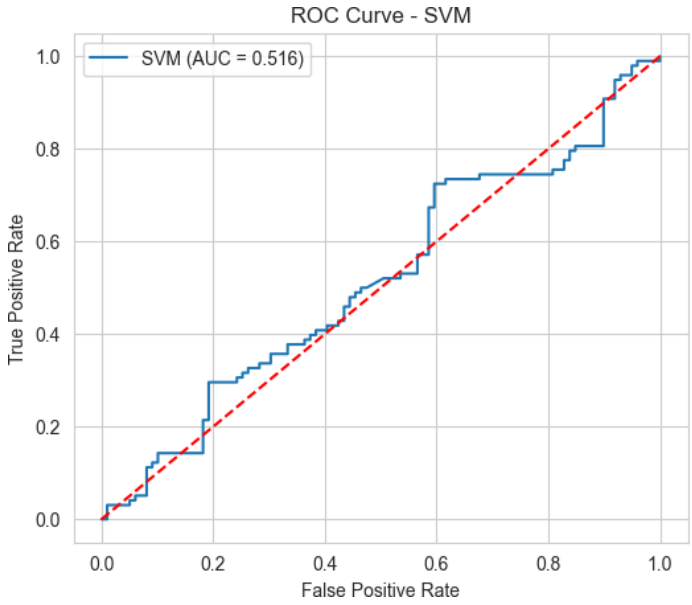

Fig. 06. ROC Curve for Support Vector Machine

1.4 Models Misclassification Summary

The fraud detection system was generated by comparing three classification models which are Logistic Regression, Random Forest, and Support Vector Machine (SVM), with a main focus on misclassifications and false negatives, as missing a fraudulent transaction is the most serious error in real-world financial systems. Logistic Regression provided the fewest overall misclassifications (11) and missed 10 fraudulent transactions, making it the most dependable model of the three. Random Forest performed slightly worse with 13 misclassifications and 11 false negatives, demonstrating no substantial improvement despite its increased complexity. In contrast, the SVM model performed badly, resulting in 88 misclassifications and missing 49 fraudulent cases, rendering it inappropriate for fraud detection in this context. Based on these findings, Logistic Regression was chosen as the final model for its balanced performance, low fraud miss rate, and consistent forecasts. The resulting model assessed the provided test transaction as normal and valid, demonstrating its practical applicability to real-world fraud detection scenarios.

6. Conclusion

This project focuses on detecting fraudulent transactions using machine learning models. Support Vector machine, Random Forest and Logistic Regression models were assessed on a balanced dataset with different metrics of performance. The outcome indicates that Logistic Regression perform very well in terms of precision and F1-score, which makes it a more appropriate model for detecting Fraud. On the other side, Random Forest also acts competitively but indicates an exchange across recall and precision, while SVM fails to distinguish fraud and non-fraud transactions effectively beneath the given setup.

Even with promising results, this research has some limitations. We make the dataset balanced by an undersampling process that minimizes the quantity of available training data and may affect the generalization of the model. Future work could identify the latest techniques to tackle imbalanced datasets, such as ensemble methods, cost-sensitive methods and SMOTE. Moreover, testing the algorithm in a real-time environment or on a big dataset could further advance practical applicability and strength.

7. References

[1] N. Baisholan, J. E. Dietz, S. Gnatyuk, M. Turdalyuly, E. T. Matson, and K. Baisholanova, "A Systematic Review of Machine Learning in Credit card fraud detection under original class imbalance," *Computers*, vol. 14, no. 10, p. 437, Oct. 2025, doi: 10.3390/computers14100437.

[2] N. Baisholan, J. E. Dietz, S. Gnatyuk, M. Turdalyuly, E. T. Matson, and K. Baisholanova, "FraUdX AI: An interpretable machine learning framework for credit card fraud detection on imbalanced datasets," *Computers*, vol. 14, no. 4, p. 120, Mar. 2025, doi: 10.3390/computers14040120.

[3] EBA and ECB, "2024 REPORT ON PAYMENT FRAUD," 2024. [Online]. Available: https://www.ecb.europa.eu/press/intro/publications/pdf/ecb.ebaecb202408.en.pdf

[4] J. K. Afriyie *et al.*, "A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions," *Decision Analytics Journal*, vol. 6, p. 100163, Jan. 2023, doi: 10.1016/j.dajour.2023.100163.

[5] S. Kumar, V. K. Gunjan, M. D. Ansari, and R. Pathak, "Credit card fraud detection using support vector machine," in *Lecture notes in networks and systems*, 2022, pp. 27–37. doi: 10.1007/978-981-16-6407-6_3.

[6] H. Z. Alenzi, N. O. Aljehane, and Department of Computer Science Tabuk University, "Fraud Detection in Credit Cards using Logistic Regression," journal-article, 2020. [Online]. Available: https://thesai.org/Downloads/Volume11No12/Paper_65-Fraud_Detection_in_Credit_Cards.pdf

[7] X. Niu, L. Wang, and X. Yang, "A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised." Apr. 24, 2019. doi: 10.48550/arxiv.1904.10604.