Online Payment Fraud Detection Model

Group 3

Machine Learning 2 (BIA – 5402)

Submitted To

Dr. Ameera Al-Karkhi

Professor,

Business Insights and Analytics,

Humber College

Designed & Built By

Pravina Prajapati - N01579926

Vrajkumar Patel – N01581006

Param Panchal – N01579822

Dhairya Dangi – N01580705

Prappan Batra – N01579150

Submitted on:

August 6, 2024

Abstract

Online payment fraud happens when someone illegally obtains another person's payment information and uses it to make unauthorized transactions or purchases. This project report presents a step-by-step data science approach for detecting fraudulent online payment transactions. The project utilizes a synthetic dataset [12][14], which simulates a series of payment transactions. The report provides a detailed account of the dataset, followed by in-depth explanations of exploratory data analysis, data cleaning, transformation, feature generation, feature selection, and model development. It also features a comparative analysis of four anomaly detection algorithms, evaluating their performance metrics after being trained on the processed dataset. The report concludes by outlining a dynamic solution for detecting online payment fraud.

***Keywords**:* Online Frauds, Fraud Detection, Anomaly Detection, Data Wrangling, Feature Generation

1. Introduction

1.1 Motivation

As online transactions become more popular across platforms like e-commerce, mobile payments, and online banking, the financial industry is growing increasingly concerned about the prevalence and prevention of online payment fraud. Payment fraud generally falls into three categories: conventional fraud (such as stolen, fake, or counterfeit cards), online fraud (such as fraudulent merchant sites), and merchant-related fraud (including merchant collusion and triangulation schemes) [9]. The US is the country most susceptible to fraud, with 34% of consumers reporting they were likely victims of fraud—a percentage that is likely even higher today [5]. According to an estimate by Juniper Research, online payment theft could result in a staggering loss of over $200 billion between 2020 and 2024, underscoring the critical need for effective fraud detection systems [8]. Moreover, the research revealed that to effectively combat increasing fraud, fraud prevention vendors need to strategically implement the right combination of verification tools at the most critical points in the customer journey. However, achieving this will require substantial capabilities [8].

Conclusive statement: The aforementioned statistics highlight the need for a dynamic approach to fraud detection that combines data science concepts with machine learning techniques to effectively address the frequent occurrence of online fraud.

1.2 Literature Survey

The available literature related to detection of online frauds using artificial intelligence points to a fact that the researchers classify this challenge in the domain of anomaly detection. [1] uses the clustering method to categorize cardholders into high, medium, and low transaction amount groups through range partitioning, and then apply the Sliding-Window method to aggregate transactions into these groups and extract features from the window to identify cardholders' behavioral patterns. [4] utilizes a random forest approach by combining around 100 decision tree classifiers for detecting anomalous behaviour. [6] compares various machine learning algorithms, including Logistic Regression, Decision Trees, and Random Forest, to determine which one performs best and could be adopted by credit card merchants for identifying fraudulent transactions. [7] utilizes a data mining and machine learning tool named WEKA in tandem with data preprocessing techniques. [10] compares the achievement of various algorithms such as Random Forest, Naïve Bayesian, Logistic Regression, Sector Vector Machine, kN, Decision Trees and GBM. All of these papers [1][4][6][7][10] utilize a time series-based transactions' dataset for their analytics and model. Distinct data mining techniques were applied in these research papers. [6] executes random sampling and feature selection over the dataset before proceeding to modelling phase. [1] aggregates the transactions into groups based on their transaction's amounts. [7] uses the concept of 10-fold cross validation as validation methodology.

2. A closer look: Online Payment Frauds Dataset

2.1 Data Excavation

The dataset utilized for generating a balanced solution to the challenge of fraud detection was borrowed from [12]. The account holder of the dataset uploaded on [12] had an aim of identifying online payment fraud using machine learning techniques. However, one of the discussions [13] posted on [12]'s Kaggle account revealed that the dataset was originally created by [14]. [14] is the original fetcher of the utilized dataset in this project. [14] introduces a synthetic dataset created with the PaySim simulator as a solution to this problem. PaySim utilizes aggregated data from a private dataset to generate a synthetic dataset that mirrors typical transaction activity while incorporating malicious behaviors [14]. Therefore, the entire understanding of the dataset following this statement has been executed based on the information provided by [14].

2.2 Schema & First Look Characteristics

The utilized dataset had the following characteristics as per [14] (the original data fetcher):

TABLE I. CHARACTERISTICS OF DATA

Data Parameter	Value
Byte size (while downloading)	186 MB
Number of rows (First look)	6,362,620 (Around 6.36 million records)
Number of columns (First look)	11
Target Column (As mentioned by [12] and [14])	isFraud
Type of data	Synthetic (generated artificially)

The schema of the dataset along with the description of columns as per [14] is as follows:

TABLE II. DATA DESCRIPTION

Column Name	DataType	Description
step	Int64	Each "step" corresponds to one hour of real-world time. The dataset spans a total of 744 steps, equivalent to a 30-day simulation period
type	String	Type of online transaction: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER
amount	Float64	The transaction's total amount
nameOrig	String	The customer initiating the transaction
oldbalanceOrg	Float64	Balance before the transaction
newbalanceOrig	Float64	Balance after the transaction
nameDest	String	Recipient of the transaction
oldbalanceDest	Float64	The recipient's starting balance prior to the transaction
newbalanceDest	Float64	The recipient's updated balance following the transaction
isFraud	Int64	Target attribute: This is the transactions made by the fraudulent agents inside the simulation.
isFlaggedFraud	Int64	The goal of the business model is to prevent large-scale transfers between accounts and to identify any unauthorized attempts.

3. Proposed Solution

3.1 Executed Ideation

Various insights were fished out via the aforementioned research papers and links. The primary focus of every research paper was to go for a pure machine learning based approach. The focus on data pre-processing was kept minimal and major limelight of research was thrown on prediction algorithms. So, there was an idea – What if we apply the "Pareto Principle" [11] for solving this challenge? We can transfer our 80% focus on data wrangling, followed by 20% focus on results and machine learning algorithms. Fraud analytics and detection can be executed in a better way if the training data is made simple and easy to understand for the algorithms. The idea of running this project stems from the data exploration step which would involve analyzing the data w.r.t to the target attribute. This dataset is an example of anomaly detection datasets which generally have low target rates. The first look target rate of this dataset is 0.19%. Thus, it was decided to execute exploratory data analysis followed by a data cleaning process which would involve removal of unnecessary rows. This step will be followed by a feature generation process which would involve generating additional processed data columns from the existing ones which will provide a better fit to the models. The next step will involve splitting the dataset in two parts: [1] Training dataset and [2] Validation dataset. The training set will be down sampled to up the target rate to 5%. This down sampled training dataset will be exposed to 3 feature selection checks: a) Low Variance Check b) Covariance Check and c) Boruta Feature Selection Algorithm [15]. These checks are capable of filtering off the features which don't provide a good fit for model development. Having passed the aforementioned checks, the training dataset with processed features will be normalized and

utilized for training 4 anomaly detection algorithms: a) Isolation Forest [16] b) One Class SVM [17] c) Autoencoder Neural Network [18] and d) Elliptical Envelope [19]. This process has ensured the 80% focus of the process in data mining and exploratory analysis followed by 20% focus in the domain of anomaly detection algorithms.

3.2 Implementation Workflow

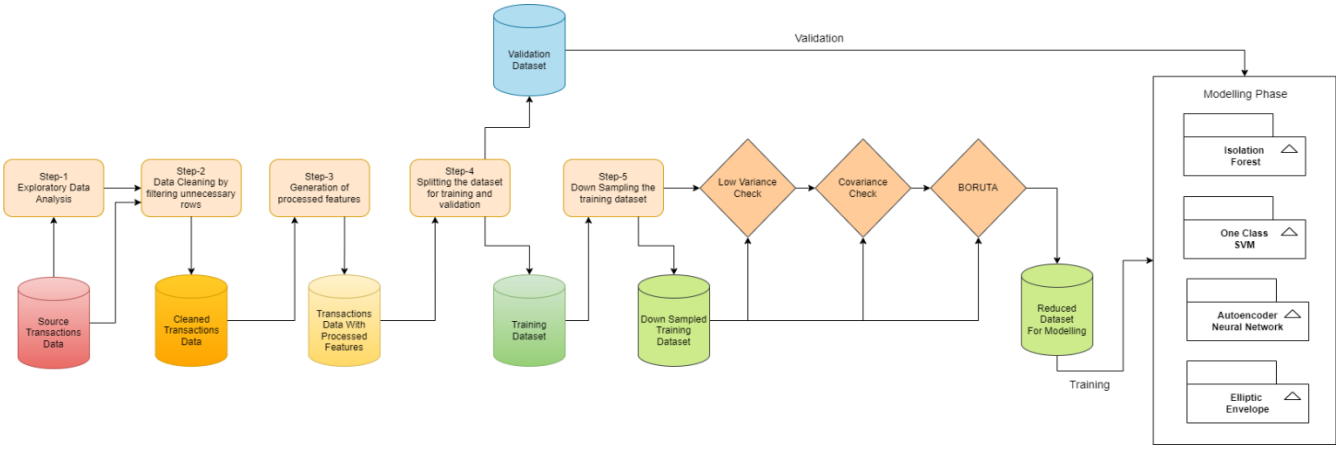

Fig. 1. Workflow Diagram

4. Exploratory Data Analysis

The first step in our journey involved analyzing the data to its core with respect to the target attribute. The utilized dataset had "isFraud" column which stores a series of target variables. However, [14] mentions that the column "isFlaggedFraud" needs to be focused upon first as it contains the transactions which were cancelled. The documentation says that these transactions should not be considered for further analytics. Removal of these 16 rows also removes the 16 positive target samples from the dataset. The next step was analyzing the "isFraud" column with respect to various parameters. The results of executing EDA over the "isFraud" column are as follows:

TABLE III. STATISTICS OF DATA

Comparison Parameter	isFraud=0	isFraud=1
Total number of rows (count of transactions)	6,354,407 (Around 6.3 M)	8197
Unique values in TYPE column	PAYMENT, DEBIT, CASH_OUT, TRANSFER, CASH_IN	TRANSFER, CASH_OUT
Average value of executed transactions amount	178, 197.04172740763 (Around 180 K)	1,461,343.157758936
Coefficient of variance of transaction amounts	3.345942085050187	1.6403036828832473
Average value of oldbalanceOrg column	832828.7117272632	1637627.6859241184
Coefficient of variance of oldbalanceOrg column	3.4666720655500334	2.1543965997717454
Average value of newbalanceOrg column	855970.2281088118	177508.20801268757
Coefficient of variance of newbalanceOrg column	3.4171596962105553	10.790362135648685

Average value of oldbalanceDest column	1101420.8745693793	545311.9582115408
Coefficient of variance of oldbalanceDest column	3.4171596962105553	6.124181220726873
Average value of newbalanceDest column	1224925.6845631592	1282205.5214859096
Coefficient of variance of newbalanceDest column	2.9992151819666044	3.051165030899726
Number of Merchants	2151495	0

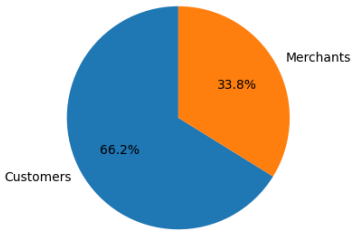

Fig. 2. Distribution of transactor type

a) Merchants were only found at destination (amount receiver's end)
b) Majority of transactions – around 66% involved customers
c) Every transaction involving merchants was found to be legitimate.

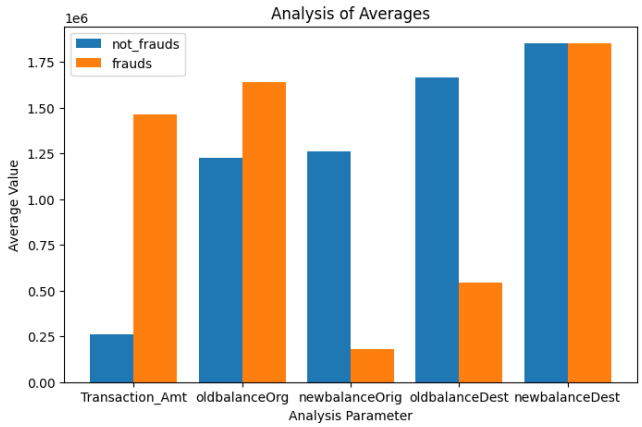

Fig. 3. Comparison of Averages

a) The average transaction amount in case of fraudulent cases was 8 times as compared to fraudulent cases.
b) The old balance at origin was more in fraudulent cases. This means that fraud cases are likely to occur more with accounts having higher balances.
c) The balance at origin after the transaction was way less in case of frauds. This means fraudulent transactions suck up majority of balances at origin.

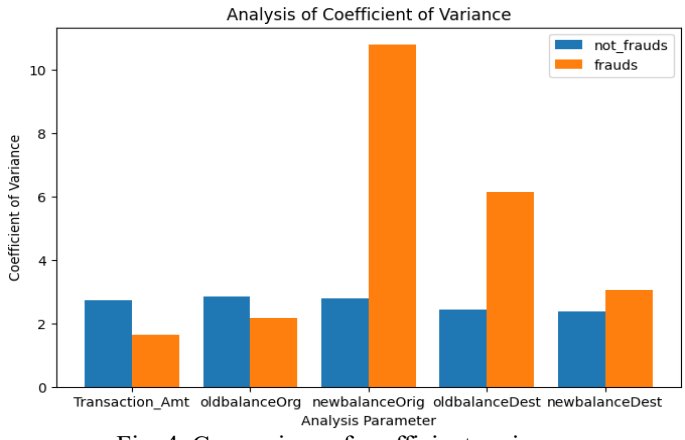

Fig. 4. Comparison of coefficient variance

a) The coefficient of variance of new balance at origin is way more in fraudulent cases. This means the fraudulent transactions cause the source balance to drain to a great extent.

b) The coefficient of variance at destination end was found more in cases of fraudulent cases. This means the fraudulent transactions increase the balance at the destination to a higher extent.

5. Data Wrangling

5.1 Data Cleaning

The insights obtained from executed EDA steps were aggregated and a forward strategy of processing the data via cleaning and transformation was decided. The first step involved cleaning the data by removing the irrelevant rows with respect to the current dataset and its documentation [12][14]. The first step of data cleaning was recommended in [14] which involved removal of all transaction rows having "isFlaggedFraud" as 1. There were 16 rows such rows which were removed as the first step to data cleaning. The next step involved removing the transaction rows which had merchants at destination end. It was found via exploratory data analysis that none of the merchants had anything to do with fraudulent transaction. Thus, we removed all transaction rows involving merchants which reduced the number of non-anomalous targets (isFraud=0). Then, the string columns of the dataset except "type" were removed which were : 'step', 'nameOrig', 'destCustomerType', 'nameDest' and 'isFlaggedFraud'. This reduced the dataset to numerical columns and the target attribute (isFraud).

Fig. 5. Data Cleaning

5.2 Data Encoding & Transformation

The cleaned dataset still had a string column – "type". The "type" column was subjected to a process of label encoding where the unique values - DEBIT, CASH_OUT, TRANSFER, CASH_IN were replaced by numbers 6, 4, 2 and 8 respectively. The dataset after executing the aforementioned cleaning and encoding process was subjected to the feature generation process explained in the next section.

5.3 Feature Generation

The standard feature columns in the dataset had data related to amounts and account balances in different phases of transactions at origin and destination accounts. The next step of data wrangling involves generation of processed columns as they would provide a better fit for the anomaly detection models. Thus, three additional columns were generated by applying a difference operation between the balance columns at the origin and destination. The additional columns can be described as follows:

TABLE IV. NEW FEATURE COLUMNS

Feature	Description
diff_newbalanceDest_oldbalanceDest	Difference between newbalanceDest & oldbalanceDest
diff_newbalanceDest_newbalanceOrig	Difference between newbalanceDest & newbalanceOrig
diff_newbalanceOrig_oldbalanceOrg	Difference between newbalanceOrig & oldbalanceOrg

The dataset after applying the aforementioned steps can be viewed as follows:

TABLE V. DATA VIEW AFTER THE FEATURE SELECTION

amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	transaction_type	diff_newbalanceDest_oldbalanceDest	diff_newbalanceDest_newbalanceOrig	diff_newbalanceOrig_oldbalanceOrg
181.00	181.00	0.00	0.00	0.00	1	2	0.00	0.00	-181.00
181.00	181.00	0.00	21182.00	0.00	1	4	-21182.00	0.00	-181.00
5337.77	41720.00	36382.23	41898.00	40348.79	0	6	-1549.21	3966.56	-5337.77
9644.94	4465.00	0.00	10845.00	157982.12	0	6	147137.12	157982.12	-4465.00
229133.94	15325.00	0.00	5083.00	51513.44	0	4	46430.44	51513.44	-15325.00
...	...	...	...	...	...	...	...	...	...
339682.13	339682.13	0.00	0.00	339682.13	1	4	339682.13	339682.13	-339682.13
6311409.28	6311409.28	0.00	0.00	0.00	1	2	0.00	0.00	-6311409.28
6311409.28	6311409.28	0.00	68488.84	6379898.11	1	4	6311409.27	6379898.11	-6311409.28
850002.52	850002.52	0.00	0.00	0.00	1	2	0.00	0.00	-850002.52
850002.52	850002.52	0.00	6510099.11	7360101.63	1	4	850002.52	7360101.63	-850002.52

5.4 Data Splitting & Down Sampling

The dataset after undergoing the cleaning and transformation process had 4211109 transaction rows and 10 columns. The observed target rate with respect to "isFraud" column was just 0.19%. This is quite common in case of anomaly detection datasets in which a whole majority of transactions are legitimate in nature and the fraudulent ones lie hidden within them. The low target rate can be increased by down sampling the number of negative targets. However, it is highly advisable to split the dataset in training and validation sets first. The validation dataset should not be subjected to down sampling as the process of down sampling alters the reality of dataset. However, the training set can be down sampled to swell the target rate to the required percentage. Moreover, the models trained over the down sampled training set must be validated against the unsampled validation set in order to make sure that the model can perform well in a real-time environment in which the overall number of fraudulent transactions will be very less as compared to legitimate ones. Thus, the original dataset was subjected to a stratified split in an 80-20 fashion with 80% of rows and positive targets in training set and rest of them in the validation set. The training set was down sampled by dropping the number of negative targets to 125,000. This increased the target rate to 5% within the down-sampled training set.

TABLE VI. STATISTICS AFTER DOWN SAMPLING

Dataset	Total Number of rows	Number of 0s in target	Number of 1s in target	Target Rate
Training Set – Unsampled	3368888	3362330	6558	0.19%
Training Set – Down Sampled	131558	125000	6558	5%
Validation Set	842221	840582	1639	0.19%

6. Feature Selection

6.1 Low Variance Check

The first step to ensuring that the processed features will be a decent fit for the incoming modelling algorithms is ensuring that they have a considerable variance among their values. An acceptable variance value ensures that the data columns are covering a lot of cases and won't overfit the model by limiting its view to a narrow and closed range of data. Thus, the down sampled training dataset was inputted to the low variance filter [20] in order to ensure that they have a decent spread of data. It was observed that all data columns had passed this check. This means that PaySim had generated the data with a wide ranged spread and the distribution was kept random in nature.

6.2 Correlation Check

Correlation is a statistical measure which quantifies how one column's values are associated with increase or decrease in other columns' values. Correlation is always measured in closed intervals between -1 and 1. High correlation can make the models consider two columns as duplicate and may learn the features in a one-dimensional manner which can affect the overall performance fatally. Thus, the features of the dataset were subjected to a correlation check among themselves and a matrix storing the correlation values was generated.

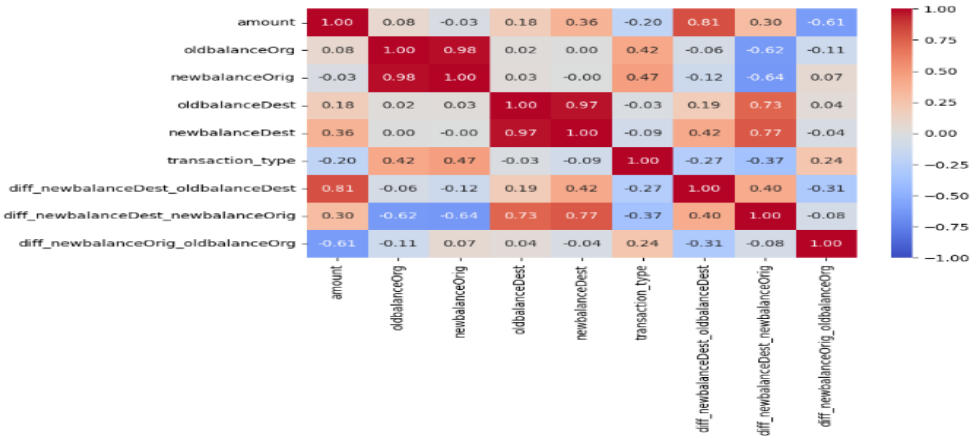

Fig. 6. Correlation matrix

The threshold of 0.85 for correlation was selected and one of the columns from pairs having correlation higher than 0.85 was removed. This resulted into removal of two columns: a) oldbalanceOrg and b) oldbalanceDest. Thus, the dataset was reduced to 6 feature columns after the correlation check.

6.3 Feature Selection Using BORUTA Algorithm

By iteratively comparing the significance of real features with randomized, noise-induced "shadow" features, the Boruta feature selection technique finds all relevant features in a dataset [15]. It functions by assigning feature importance scores after training a machine learning model, usually a random forest. Features that continuously score lower than the best-performing shadow feature are rejected as irrelevant, while those with importance scores significantly higher than the shadow feature are considered important [15]. Until the algorithm converges or reaches a predetermined number of iterations, the procedure is repeated [15]. The features were ingested to a Boruta check with 100 iterations. The implementation was done using the BorutaPy library. It was observed that all 6 features had passed the Boruta check and all of them were processed towards the next phase – modelling.

7. Modelling

A bunch of machine learning algorithms were studied along with their impact during the literature survey [1][4][6][7]. However, the focus of modelling was narrowed down to 4 anomaly detection algorithms. The models were trained based on the down sampled training dataset with the features which had passed all 3 aforementioned checks. The algorithms were executed and the trained models were tested against the unsampled validation dataset in order to ensure that they perform well in real-time situations.

7.1 Isolation Forest

An anomaly detection technique called Isolation Forest isolates data points in order to find outliers [16]. The way it operates is that features are chosen at random, and then split values between the minimum and maximum values of those features are also chosen at random [16]. The data is recursively partitioned in this manner, resulting in a tree structure where sparse anomalies are more likely to be isolated early on. Anomalies have a shorter path length from the root to the point of isolation than regular points [16]. Based on the average path length over a large number of trees, the algorithm determines an anomaly score. It is thought that points with shorter average journey lengths are more unusual. Effective and well-suited to high-dimensional data, Isolation Forest [16].

7.2 One Class Support Vector Machine

An unsupervised learning technique called One-Class SVM (Support Vector Machine) is employed in anomaly detection [17]. By identifying a decision boundary in a high-dimensional space that most effectively captures the bulk of the training data points, it models the "normal" data [17]. Anomalies are defined as points that are outside of this line. Using a kernel function, the algorithm converts the supplied data into a feature space with higher dimensions, where it aims to maximize the margin around the data cluster [17]. When there are few to no anomalies and

a large percentage of typical samples in the training data, it works especially well. In situations such as fraud detection and outlier detection, One-Class SVM is frequently utilized [17].

7.3 Auto Encoder Neural Networks

By iteratively comparing the significance of real features with randomized, noise-induced "shadow" features, the Boruta feature selection technique finds all relevant features in a dataset [15]. It functions by assigning feature importance scores after training a machine learning model, usually a random forest. Features that continuously score lower than the best-performing shadow feature are rejected as irrelevant, while those with importance scores significantly higher than the shadow feature are considered important [15]. Until the algorithm converges or reaches a predetermined number of iterations, the procedure is repeated [15]. To accommodate alternative data types, such as pictures and time series, the architecture can be modified by adding more layers or changing the activation function [18]. Autoencoders are valuable in a variety of machine learning applications because they are strong instruments for identifying intricate patterns and structures in data.

7.4 Elliptic Envelope

Elliptic Envelope is an anomaly detection approach that models the data with a multivariate normal distribution and assumes that it follows a Gaussian distribution. In order to represent the underlying data distribution, it fits an ellipsoid to the central data points. Anomalous data points are those that fall outside of this ellipsoid. The ellipsoid's form and orientation are determined by the method utilizing an estimate of the data's mean and covariance. Elliptic Envelope may have trouble with complex or non-Gaussian distributions, but it works well when the data is roughly Gaussian. When dealing with multivariate data, such financial or environmental datasets, it is especially helpful in identifying outliers.

7.5 Performance Metrics

The following performance metrics were generated from testing the performance of the aforementioned models on the unsampled validation data:

TABLE VI. COMPARISION OF PERFORMANCE MATRIX

Model	Accuracy	Precision	Recall	AUC
Isolation Forest	95.77%	1.2%	26.36 %	61.13 %
One Class SVM	95.38%	0.7%	17.88 %	56.70 %
AutoEncoders	99.36%	0.19%	100%	50%
Elliptic Envelope	**92.47 %**	**59.25 %**	**87.73 %**	**81.29 %**

The aforementioned statistics confirm that Elliptic Envelope Algorithm performs the best when it comes to accuracy, precision, recall and AUC. The isolation forest fails to identify the positive samples as the precision is very low. Autoencoders have clearly under fitted as all transactions in validation set were classified as legitimate. The One class SVM fails to identify a lot of fraudulent transactions. The Gaussian properties of Elliptic Envelope seem to fit the fraudulent transactions.

8. Conclusion

The analysis and modelling have proven that the dataset provided by [12][14] can be processed using an elliptic envelope model for analyzing a high volume of fraudulent transactions. However, we believe that purely relying on a machine learning based approach may cause the accuracy to degrade gradually due to data drifts. Thus, constant monitoring of the performance metrics of model is required and the model needs to be refined over new datasets in case of drifts. Moreover, we can increase the confidence of determining fraudulent transactions based on certain observations from the exploratory data analysis phase. An example can be the sequence of fraudulent transactions. Every fraudulent transaction has occurred with a consecutive dependency on a previous transaction. The “Type” column in case of fraudulent transactions carries only two unique values: TRANSFER and CASH OUT. Thus, if the check of fraudulent transactions using machine learning can be just limited to the sequence of transactions of these two types. These conditions can be manually added apart from the results in order to be perfectly sure if an incoming transaction is fraudulent or not.

9. References

[1] Dornadula, V.N. and S, G. (2019) 'Credit Card Fraud Detection using Machine Learning Algorithms', INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING [Preprint].

[2] 60+ global credit card fraud statistics in 2024 Techopedia. Available at: https://www.techopedia.com/credit-card-fraud-statistics

[3] Online payment fraud 101: What it is, types, and how to prevent it (2024) Zoho Books. Available at: https://www.zoho.com/books/academy/banking-and-payments/payment-fraud.html

[4] Namani, S. et al. (2024) 'ONLINE PAYMENT FRAUD DETECTION: AN INTEGRATED APPROACH', International Research Journal of Modernization in Engineering Technology and science [Preprint]. doi:10.56726/irjmets.

[5] Mastercard, (2024) Ecommerce fraud trends and statistics merchants need to know, Payment and cybersecurity solutions. Available at: https://b2b.mastercard.com/news-and-insights/blog/ecommerce-fraud-trends-and-statistics-merchants-need-to-know-in-2024/

[6] SVSS, L. and Kavila, S.D. (2018) 'Machine Learning For Credit Card Fraud Detection System', International Journal of Applied Engineering Research [Preprint]. doi:10.37622/ijaer.

[7] Yee, O.S., Sagadevan, S. and Malim, N.H.A.H. (2018) 'Credit Card Fraud Detection Using Machine Learning As Data Mining Technique', Journal of Telecommunication, Electronic and Computer Engineering, 10. doi:e-ISSN: 2289-813.

[8] Smith, S. (2022) Online Payment Fraud Losses to Exceed $343 Billion Globally Over the Next 5 Years, juniperresearch. Available at: https://www.juniperresearch.com/press/online-payment-fraud-losses-to-exceed-343bn/

[9] Bhatla, T.P., Prabhu, V., and Dua, A. (2003). understanding credit card frauds. Crads Business Review# 2003-1, Tata Consultancy Services.

[10] Simaiya, S. et al. (2020) 'An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods', MATTER International Journal of Science and Technology [Preprint]. doi:10.20319/mijst.

[11] Team, T.I. What is the pareto principle-aka the pareto rule or 80/20 rule?, Investopedia. Available at: https://www.investopedia.com/terms/p/paretoprinciple.asp

[12] Shah, J. (2022) Online payment fraud detection, Kaggle. Available at: https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection

[13] Shah, J. (2022) Discussions - Online payment fraud detection, Kaggle. Available at: https://www.kaggle.com/datasets/jainilcoder/online-payment-fraud-detection

[14] Lopez-Rojas, E. (2017) Synthetic financial datasets for fraud detection, Kaggle. Available at: https://www.kaggle.com/datasets/ealaxi/paysim1/data

[15] Jankowski, A. and Rudnicki, W. (2010) 'Boruta - A System for Feature Selection', Fundamenta Informaticae [Preprint]. doi:10.3233/FI-2010-288.

[16] Xu, H. et al. (2023) 'Deep Isolation Forest for Anomaly Detection', IEEE Transactions on Knowledge and Data Engineering [Preprint].

[17] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," Pattern Recognition, vol. 58, pp. 121-134, 2016. [Online]. Available: https://doi.org/10.1016/j.patcog.2016.03.028.

[18] A. Legrand, H. Trannois, and A. Cournier, "Use of Uncertainty with Autoencoder Neural Networks for Anomaly Detection," in 2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), Sardinia, Italy, Jun. 2019, pp. 32-35. doi: 10.1109/AIKE.2019.00014.

[19] L. Forzani and Z. Su, "Envelopes for Elliptical Multivariate Linear Regression," Statistica Sinica, vol. 31, no. 1, pp. 301-332, 2021. [Online]. Available: https://www3.stat.sinica.edu.tw/statistica/J31N1/J31N113/J31N113.html

[20] Analytics Vidhya. "Beginner's Guide to Low Variance Filter and Its Implementation," Analytics Vidhya, Apr. 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/04/beginners-guide-to-low-variance-filter-and-its-implementation/