Isolation Forests as a Solution to Credit Card Fraud

C. Dilger

Artificial Intelligence Management at School of Business in Farmingdale State College 2350 NY-110, Farmingdale, NY 11735

First published October 22, 2025. Current version published October 22, 2025.

Abstract: Isolation Forests were compared against a range of alternative options for the implementation of credit card fraud. It is reviewed how Isolation Forests compete against Local Outlier Factor, One Class SVM, and a deep autoencoder. Experimentation is done to test Isolation Forests against LightGBM. An overall analysis of Isolation Forests against alternative options is completed.

Index Terms: Fraud, Artificial Intelligence.

1. Introduction

Credit card fraud is a significant issue in the realm of economics. According to the statistics, two-thirds of U.S. credit holders have experienced fraud, with just over half experiencing it multiple times. This affects tens of millions of Americans each year, exceeding $6.2 billion in costs last year alone [1]. The statistics demonstrate how widespread and expensive an issue this is in the U.S. It is an epidemic, and preventative steps are required in order to keep the issue under control.

This is where Artificial Intelligence, or AI, tends to become useful. AI has the power to perform both supervised and unsupervised classification of data. If that data is a transaction, AI can be employed to classify between a safe transaction and a fraudulent transaction, and with a decent amount of accuracy. With these methods, it may be possible to detect fraud as it happens and immediately put a halt to it.

The most common type of algorithm for detecting fraud is outlier detection. Since most transactions made by a user throughout their lifetime are non-fraudulent, fraudulent transactions are rare, often one-off instances where a transaction is made from someone other than the user. In other words, fraudulent transactions are outliers. They stand out among the many transactions made by a user. Therefore, we want algorithms equipped with the ability to detect outliers as they occur.

There are many algorithms that perform the role of detecting outliers. Among the most notorious, and perhaps the most effective, is the Isolation Forest. Developed by Liu, Ting, and Zhou in 2008, the Isolation Forest was introduced as a powerful tool fitted for outlier detection. The idea of Isolation Forests is fundamentally simple; when putting data through a tree, outliers are prone to ”isolation.”

Isolation Forests are a series of Isolation Trees, or iTrees. These are binary tree structures built through a random, recursive partitioning of data. At each node, an attribute is selected, and data points are split based on their values for this attribute. The process is repeated until the tree is completed. The tree completes when either all data is isolated or a predetermined height is

reached. Since outliers have their own unique values that do not correlate with other data points, they are easily isolated. Therefore, outliers require less partitions to be isolated. As a result, outliers tend to isolate closer to the root of the tree [2].

The number of edges from root to termination is measured by the path length, $h(x)$. However, Isolation Forests are a series of iTrees, not one lone iTree. Therefore, we require an ensemble of trees, computing the average path length $E(h(x))$. This is normalized by an anomaly score, $s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$ where $c(n) = 2H(n-1) - \frac{2(n-1)}{n}$ represents the average path length in binary search trees. H(i) is the harmonic number. Anomalies are indicated by nearing a score of 1.0, and scores below 0.5 represent normal values.

Isolation Forests also have great efficiency. For training, there is a time complexity of $(O(t\psi \log \psi))$. For evaluation, there is a time complexity of $(O(nt \log \psi))$. Additionally, in testing, Isolation Forests outperform methods such as ORCA, LOF, and Random Forests. This is an undeniably powerful method. Throughout this document, we will be comparing it against alternative methods.

2. Related Work

Isolation Forests have undergone rigorous testing throughout their existence. In this section, we will go over the existing works that aim to test the Isolation Forest against alternative methods.

2.1. Isolation Forests

This testing goes back as far as the original document where Isolation Forests were proposed. In the original document, titled Isolation Forests, the algorithm is compared to ORCA, LOF, and Random Forests. According to the document, "Our empirical evaluation shows that iForest performs favourably to ORCA, a near-linear time complexity distance-based method, LOF and Random Forests in terms of AUC and processing time, and especially in large data sets" [2]. The overall results can be seen in Table I.

	AUC				Time (seconds)					
					iForest					
	iForest	ORCA	LOF	RF	Train	Eval.	Total	ORCA	LOF	RF
Http (KDDCUP99)	**1.00**	0.36	NA	NA	0.25	15.33	15.58	9487.47	NA	NA
ForestCover	0.88	0.83	NA	NA	0.76	15.57	16.33	6995.17	NA	NA
Mulcross	0.97	0.33	NA	NA	0.26	12.26	12.52	2512.20	NA	NA
Smtp (KDDCUP99)	0.88	0.80	NA	NA	0.14	2.58	2.72	267.45	NA	NA
Shuttle	**1.00**	0.60	0.55	NA	0.30	2.83	3.13	156.66	7489.74	NA
Mammography	0.86	0.77	0.67	NA	0.16	0.50	0.66	4.49	14647.00	NA
Annthyroid	0.82	0.68	0.72	NA	0.15	0.36	0.51	2.32	72.02	NA
Satellite	0.71	0.65	0.52	NA	0.46	1.17	1.63	8.51	217.39	NA
Pima	0.67	0.71	0.49	0.65	0.17	0.11	0.28	**0.06**	1.14	4.98
Breastw	0.99	0.98	0.37	0.97	0.17	0.11	0.28	**0.04**	1.77	3.10
Arrhythmia	0.80	0.78	0.73	0.60	2.12	0.86	2.98	0.49	6.35	2.32
Ionosphere	0.85	**0.92**	0.89	0.85	0.33	0.15	0.48	**0.04**	0.64	0.83

TABLE I
COMPARISON OF AUC SCORES AND EXECUTION TIMES ACROSS DIFFERENT DATASETS AND METHODS.

Table I provides a lot of context into the performance of Isolation Forests against ORCA, LOF, and Random Forests. Notably, iForest performs better than ORCA, particularly with datasets where $n > 1000$. iForests also performs much faster, especially when $n > 1000$. There are gaps in the table because LOF has too high computational complexity for some datasets, and RF has too large of a memory requirement. This contributes a number of comparisons already, however there is many more to cover.

2.2. Advanced fraud detection using machine learning models: enhancing financial transaction security

Isolation Forests alone did not put Isolation Forests against all possible outlier detection algorithms. In fact, it is far from it. Fortunately, there are more examples of comparisons between Isolation

Forests and alternative options. This takes us to the document, *Advanced fraud detection using machine learning models: enhancing financial transaction security*. The goal of this document is to solve the issue of detecting credit card fraud with the implementation of 3 algorithms: Isolation Forests, One Class SVM, and a deep autoencoder.

This means that we get a thorough comparison between the three algorithms, and additional information on how they can be used to solve our main issue — credit card fraud. After rigorous testing, the researchers came up with the results shown in Table II [3].

Model	Detection Rate (%)	False Positive Rate (%)	Precision (%)	AUC-ROC
Isolation Forest	95.3	4.8	91.6	0.964
One-Class SVM	95.0	5.1	89.9	0.958
Autoencoder (AE)	94.7	4.5	92.3	0.971
K-Means (k=3)	N/A (clustering only)	N/A	N/A	N/A
DBSCAN	N/A (clustering only)	N/A	N/A	N/A

TABLE II
EVALUATION RESULTS SUMMARY

These results are informative. Isolation Forests clearly have the highest detection rate of the three algorithms. However, it actually underperformed in precision and AUC against the autoencoder. This presents autoencoders as a particularly viable alternative to Isolation Forests, though the differences in performance are minimal as both methods perform excellently. While this provided important information on how Isolation Forests perform against a number of algortihms, it is still not comprehensive. There are many more algorithms to test against Isolation Forests.

2.3. Credit Card Fraud Detection using Local Outlier Factor and Isolation Forest

So far, all tests showed Isolation Forests to be highly competitive. This is not a universal phenomenon, however. There are times when Isolation Forests fail to meet expectations, so it's critical to select the correct algorithm for your data. One example of testing where Isolation Forests underperformed was *Credit Card Fraud Detection using Local Outlier Factor and Isolation Forest*. This document aims to compare Isolation Forests to Local Outlier Factor specifically in our case of detecting credit card fraud, and makes several additional comparisons in the process. Isolation Forests are compared to Local Outlier Factor, Logistic Regression, Decision Tree, and Random Forests. Isolation Forests are found to be the worst performer in accuracy, while Local Outlier Factor is found to be the best performer. The overall results can be found in Table III [4].

Algorithm	Accuracy
Logistic Regression	90.0%
Decision Tree	94.3%
Random Forest Classifier	95.5%
Isolation Forest	71%
Local Outlier Factor	97%

TABLE III
COMPARISON OF VARIOUS ALGORITHMS.

This presents an issue. While Isolation Forests are proven in the original Isolation Forests paper to *generally* outperform LOF, that is not fully conclusive. Results may vary based on the database, and the credit card fraud dataset is one example of a dataset where Isolation Forests do not perform.

While this document provided context in how Isolation Forests perform against many algorithms in the case of credit card fraud, it is still not comprehensive. There are still algorithms to compare to Isolation Forests in the context of credit card fraud.

2.4. Using Isolation Forest in anomaly detection: the case of credit card transactions

Still, while Isolation Forests had a failure in the last paper, performance heavily weighs not solely on the topic, but on the dataset used. It may still be possible to have a credit card fraud database, and have Isolation Forests as the top performer, as is the case with this next document. In the document *Using Isolation Forest in anomaly detection: the case of credit card transactions*, Isolation Forests are once again tested in the capacity to detect credit card fraud. Once again, it was put against several contestants — OCSVM, LOF, and K-Means.

The hypothesis was that Isolation Forests would prevail. According to the paper, "Many advantages make Isolation forest outrank other methods in anomaly detection algorithm. First, it only needs small samples from large datasets so as to derive an anomaly detection function which makes it fast and scalable. Second, it does not require example anomalies in the training dataset. Third, the tree depth is the basis of its distance threshold for determining anomalies which is autonomous from the scaling of the dataset dimensions" [5]

In other words, Isolation Forests are an immensely powerful tool that provide a number of benefits into the field of anomaly detection. The performance demonstrates this, as well. Table IV provides the documentation for how the three algorithms performed.

Methods	F1 Score	Accuracy	AUC Score
OCSVM	0.0033	0.5088	0.5154
LOF	0.0027	0.8901	0.4970
K-means	0.0054	0.9012	0.5191
Isolation Forest	0.0544	0.9512	0.9168

TABLE IV
COMPARISON OF DIFFERENT ANOMALY DETECTION METHODS.

Isolation Forests had the highest F1 Score, Accuracy, and AUC Score out of the four algorithms. Does this mean that Isolation Forests are the best, and the tests that said otherwise were a fluke? There's more to test, but the evidence thus far suggests that Isolation Forests are generally the best, with struggles in some datasets

2.5. Comparative Evaluation of Anomaly Detection Methods for Fraud Detection in Online Credit Card Payments

.

At this point, we've compared Isolation Forests to OCSVM, LOF, K-Means, Autoencoders, ORCA, and Random Forests. There are still plenty more models to compare Isolation Forests to. Particularly, LightGBM has substantial potential. The article *Comparative Evaluation of Anomaly Detection Methods for Fraud Detection in Online Credit Card Payments* addresses this comparison in the context of credit card fraud, once again. In the article, Isolation Forests are compared to LightGBM, ECOD, COPOD, KNN, GOAD, NeuTraL-AD, Internal Control, and NPT-AD. This is the most comprehensive comparison yet. The results of the comparison can be found in Table V.

This time, LightGBM was the top performer. Isolation Forests were a powerful contender, but LightBGM was significantly better. On almost every metric, LightGBM performed with more accuracy than Isolation Forests. No other model was as consistent in defeating Isolation Forests.

3. Methods

Throughout the related works, Isolation Forests have defeated nearly every alternative method of anomaly detection tested. This is the case across all methods aside from LightGBM, which defeated Isolation Forests across all metrics in the one paper that compared them. The goal of this implementation will be to rigorously test Isolation Forests against LightGBM to see if LightGBM is truly a more accurate model.

Model	Country A						Country B					
	F1 (↑)		AUROC (↑)		AUPRC (↑)		F1 (↑)		AUROC (↑)		AUPRC (↑)	
	2018	2020	2018	2020	2018	2020	2018	2020	2018	2020	2018	2020
LightGBM	21.52	0.7	89.98	66.49	18.74	0.31	17.15	0.48	93.5	75.51	34.84	2.68
	1.6	0.3	0.1	2.38	1.78	0.05	1.93	0.39	0.31	1.24	1.48	0.29
ECOD	0.48	0.2	62.2	62.49	0.4	0.25	0.57	1.04	54.02	51.59	1.09	0.76
	0.2	0.22	0.64	1.02	0.02	0.01	0.26	0.38	0.54	0.87	0.06	0.04
COPOD	0.34	0.16	64.77	62.64	0.43	0.25	0.5	1.12	51.7	50.15	1.0	0.73
	0.21	0.15	0.51	0.98	0.02	0.01	0.2	0.44	0.59	0.91	0.05	0.04
Isolation Forest	0.16	0.19	64.14	60.55	0.43	0.23	0.71	1.3	60.52	46.86	1.35	0.67
	0.12	0.19	0.75	0.76	0.02	0.01	0.23	0.34	0.52	0.84	0.07	0.03
KNN	0.34	0.01	68.87	55.6	0.51	0.18	0.78	0.38	65.92	49.28	1.58	0.63
	0.12	0.04	0.76	0.85	0.02	0.01	0.21	0.27	0.64	0.67	0.08	0.02
GOAD	0.14	0.19	53.72	52.73	0.17	0.17	0.7	0.69	50.36	64.45	0.67	1.03
	0.09	0.13	1.41	1.69	0.01	0.01	0.36	0.34	2.35	1.25	0.05	0.06
NeuTraL-AD	0.6	0.02	59.12	51.52	0.35	0.15	1.45	0.38	53.23	45.19	1.08	0.58
	0.22	0.08	3.56	1.15	0.05	0.01	0.44	0.21	1.9	1.75	0.07	0.03
Internal Cont.	0.64	0.0	39.43	46.7	0.18	0.13	1.21	0.23	45.63	50.66	0.87	0.68
	0.08	0.0	1.05	0.17	0.0	0.0	0.16	0.07	2.46	0.9	0.1	0.03
NPT-AD	0.97	0.66	67.21	53.2	0.81	0.18	1.67	0.58	66.21	53.45	1.28	0.61
	0.07	0.06	1.25	0.65	0.01	0.03	0.11	0.03	1.14	0.43	0.11	0.06

TABLE V
PERFORMANCE COMPARISON ACROSS COUNTRY A AND COUNTRY B FOR YEARS 2018 AND 2020.

3.1. Isolation Forests

In order to achieve this, our model will start off testing out an Isolation Forest. Isolation forests take advantage of the idea that when you create a tree by splitting a dataset apart through features, anomalies will tend to become isolated within the tree closer to the root of the tree. Our primary dataset comes with 28 PCA components, labeled as V1 through V28. The dataset also contains the amount of a transaction, and the classification of a transaction, where 0 is safe and 1 is fraudulent. Our method will create t isolation trees through recursively splitting the dataset through random feature q selection and split value p.

The path length, $h(x)$, or the number of splits to isolate x, is combined with theoretical normalization by the anomaly score, given by $s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}}$, where n is the sample size and $E[h(x)]$ is the average path length across trees. The harmonic number, $H(n-1)$, normalizes $c(n) = 2H(n-1) - 2(n-1)/n$. Scores that near 1 indicate fraud, and scores near 0.5 indicate a safe transaction.

For example, transaction x is defined by the data (v1: 1.449, v2: -1.176, ..., v28: 0.016, Amount: 7.8, Class: 0). The result of the model is $h(x) = 10$ splits, which produces $s(x,n) = 0.48$. The result is nearing 0.5, so it is a normal transaction.

Algorithm 1 Isolation Forest

Input: Dataset $\mathbf{D}$, trees t, subsample ψ
Initialize forest $F \leftarrow \emptyset$
for $i = 1$ to t **do**
  Sample ψ transactions from $\mathbf{D}$ to form $\mathbf{D}'$
  Build tree $T_i \leftarrow$ iTree($\mathbf{D}'$, 0, limit)
  $F \leftarrow F \cup \{T_i\}$
end for
for each $\mathbf{x} \in \mathbf{D}$ **do**
  Compute $E[h(\mathbf{x})] \leftarrow$ avg PathLength$(\mathbf{x}, T_i)$ $\forall T_i \in F$
  Calculate $s(\mathbf{x}, \psi) = 2^{-E[h(\mathbf{x})]/c(\psi)}$
end for
return anomaly scores $s(\mathbf{x}, \psi)$

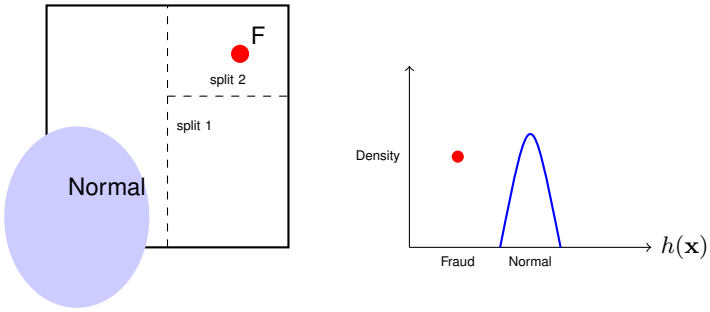


Fig. 1. Left: Random partitioning isolates fraud (F) faster than normal transactions. Right: Path length distribution shows fraud has shorter paths.

3.2. LightGBM

LightGBM is a very different method of detecting fraud, detecting it through a gradient-based sequential ensemble learning. With the same transaction as before, transaction x, decision trees are iteratively made, minimizing classification loss by fitting residuals from previous trees. M trees are combined through additive assemble $\hat{y}(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x})$ where $f_m(x)$ is the m-th tree's output and $\hat{y}(x)$ is the fraud possibility. Each tree minimizes the loss function gradient at iteration m, $f_m = \arg\min_f \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(m-1)} + f(\mathbf{x}_i)) + \Omega(f)$, where L is the binary cross-entropy loss, $\hat{y}_i^{m-1}$ is the cumulative prediction for trees 1 to $m-1$ and $\Omega(f)$ is regularization that prevents overfitting. Going back to transaction x will produce $f_1(x) = -0.3, f_2(x) = -0.15$. This results in $\hat{y}(x) = sigmoid(-0.45) = 0.39$, which results in "normal" classification.

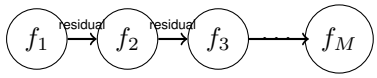
Algorithm 2 LightGBM Classifier

Input: Dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, learning rate η, trees M
Initialize $\hat{y}_i^{(0)} = 0$ for all i
for $m = 1$ to M **do**
  Compute gradients $g_i = \frac{\partial L(y_i, \hat{y}_i^{(m-1)})}{\partial \hat{y}_i^{(m-1)}}$ for all i
  Compute Hessians $h_i = \frac{\partial^2 L(y_i, \hat{y}_i^{(m-1)})}{\partial (\hat{y}_i^{(m-1)})^2}$ for all i
  Build histogram-based tree f_m fitting $-g_i$ using leaf-wise growth
  Update predictions $\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta \cdot f_m(\mathbf{x}_i)$ for all i
end for
return final predictions $\hat{y}(\mathbf{x}) = \text{sigmoid}(\hat{y}^{(M)})$

4. Results

Comparing Isolation Forests to LightGBM was a multi-step process that required several datasets and test runs. We started off with *Credit Card Fraud Detection*, a dataset available on Kaggle that contains European transactions from September 2013. The dataset was already primarily preprocessed, however it suffered from a major class imbalance. This proposed an ablation study, whether or not the class imbalance would impact the performance of either algorithm enough to change the results. We started off by running SMOTE, and then testing the performance of both models against the dataset. The ful results are shown at Table VI. However, the gist is that LightGBM greatly outperformed Isolation Forests. For example, AUC for Isolation Forests was .9652, while it was a whopping 1.0 for LightGBM. LightGBM performed almost perfectly, while Isolation Forest suffered from some inaccuracy.

We were interested in whether or not Isolation Forests would suddenly become competitive if the dataset was left imbalanced. After all, anomaly detection should not be harmed by a class



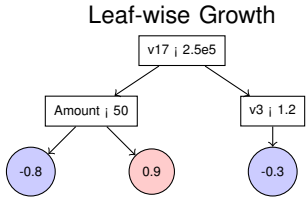


Fig. 2. Top: Sequential boosting where each tree fits residuals from previous trees. Bottom: Leaf-wise tree expansion splits the leaf with maximum loss reduction (red indicates fraud prediction).

Metric	**Isolation Forest**	**LightGBM**
Basic Performance Metrics		
Accuracy	0.9655	**0.9990**
Precision	0.6465	**0.9800**
Recall	0.6531	**1.0000**
F1-Score	0.6497	**0.9899**
ROC-AUC	0.9652	**1.0000**
MCC	0.6316	**0.9894**
Cross-Validation Analysis (5-Fold)		
Accuracy	0.9626 ± 0.0043 (95% CI: [0.9633, 0.9752])	0.9997 ± 0.0002 (95% CI: [0.9994, 1.0001])
Precision	0.6879 ± 0.0497 (95% CI: [0.6189, 0.7569])	0.9996 ± 0.0003 (95% CI: [0.9992, 1.0001])
Recall	0.6924 ± 0.0597 (95% CI: [0.6099, 0.7758])	0.9999 ± 0.0003 (95% CI: [0.9993, 1.0006])
F1-Score	0.6890 ± 0.0444 (95% CI: [0.6274, 0.7506])	0.9997 ± 0.0002 (95% CI: [0.9994, 1.0001])
ROC-AUC	0.9750 ± 0.0050 (95% CI: [0.9680, 0.9821])	1.0000 ± 0.0000 (95% CI: [1.0000, 1.0000])
Statistical Significance Tests		
T-test (Fraud vs Normal)	t-statistic: 38.5888 p-value: 6.4478e-244	t-statistic: 401.8760 p-value: 0.0000e+00
Significance at $\alpha = 0.05$	Yes	Yes
Cohen's d (Effect Size) Interpretation	3.9972 large effect	**41.6283** large effect

TABLE VI
COMPREHENSIVE COMPARISON OF ISOLATION FOREST AND LIGHTGBM PERFORMANCE WITH STATISTICAL ANALYSIS.

imbalance. It relies on one class being an anomaly. So, we ran the models again without SMOTE. Table VII contains the full statistics of this comparison. However, the results are simple — it had little effect. LightGBM still did nearly perfect, and Isolation Forests still struggled.

It was still possible that it was simply a dataset where LightGBM had an advantage. The next goal was to thoroughly test whether or not LightGBM would be consistent in its superiority over Isolation Forests. Therefore, the next goal was to test the models on another *Credit Card Fraud* dataset. This time, the columns were completely different. Instead of the 28 heavily preprocessed features that the last model had, this model had only 8 features, including the target feature. These features included distance from home, and whether or not the users pin was used. The full results are available in the table VIII. Isolation forests did not perform over .50 in any metric,

SIDE-BY-SIDE COMPARISON (WITHOUT SMOTE)		
	Isolation Forest	**LightGBM (Class Weights)**
Accuracy	0.965500	0.999000
Precision	0.646465	1.000000
Recall	0.653061	0.979592
F1-Score	0.649746	0.989691
ROC-AUC	0.965176	0.999938
MCC	0.631612	0.989223

TABLE VII
SIDE-BY-SIDE COMPARISON OF ISOLATION FOREST AND LIGHTGBM PERFORMANCE WITHOUT SMOTE.

while LightGBM performed near 1.0 on every metric.

SIDE-BY-SIDE COMPARISON – DATASET 2		
	Isolation Forest	**LightGBM**
Accuracy	0.445250	0.998000
Precision	0.445387	0.996100
Recall	0.446500	0.999500
F1-Score	0.445943	0.998003
ROC-AUC	0.484863	0.999982
MCC	-0.109500	0.996004

TABLE VIII
SIDE-BY-SIDE COMPARISON OF ISOLATION FOREST AND LIGHTGBM PERFORMANCE ON DATASET 2.

We compared it between one more dataset, the *Credit Card Fraud Detection Dataset 2023*. This had similar columns to the first dataset, but it was data collected 10 years later. We tested both models on this dataset, and got the values displayed in Table IX. In summary, Isolation Forests once again lost to LightGBM. Isolation Forests had a ROC-AUC of .647, and LightGBM had a ROC-AUC of nearly 1.0.

SIDE-BY-SIDE COMPARISON – DATASET 3		
	Isolation Forest	**LightGBM**
Accuracy	0.591250	0.999500
Precision	0.592781	0.999500
Recall	0.583000	0.999500
F1-Score	0.587850	0.999500
ROC-AUC	0.646721	0.999999
MCC	0.182525	0.999000

TABLE IX
SIDE-BY-SIDE COMPARISON OF ISOLATION FOREST AND LIGHTGBM PERFORMANCE ON DATASET 3.

Isolation Forests seem to be performing poorly. Upon investigation, it is likely that our pre-processed datasets are not making the fraud class ”rare” enough to be detected as anomalies. Therefore, we gave Isolation Forests a second chance with a 10:1 ratio of normal to fraud. The results made it clear that Isolation Forests were still not getting competitive numbers. The results are in Table X.

5. Conclusion

The anomaly detection capabilities of Isolation Forests is very effective in the classification of credit card transactions as safe or fraudulent. By performing random splits on data, the algorithm isolates each data point and classifies data points near the root as anomalies. When it comes to credit card fraud, these anomalies are fraudulent transactions. Our research has shown that in the majority of cases, iForests outperform models such as ORCA, LOF, Random Forests, One-Class SVM, Autoencoders, K-Means, and more, when used in detecting fraudulent transactions.

Isolation Forest (10% fraud)		
Metric	**Dataset 2**	**Dataset 3**
Accuracy	0.8438	0.9170
Precision	0.2137	0.5867
Recall	0.2100	0.6058
F1-Score	0.2119	0.5808
ROC-AUC	0.7276	0.9064
MCC	0.1251	0.5348

TABLE X
ISOLATION FOREST PERFORMANCE COMPARISON ON DATASET 2 AND DATASET 3 WITH 10% FRAUD RATE.

Our research also suggested that LightGBM may outperform Isolation Forests for this implementation. This was then rigorously put to the test, as iForests and LightGBM competed in 3 different fraud datasets. The results were conclusive, LightGBM consistently performed nearly flawlessly, while Isolation Forests performed relatively inconsistently and less effectively. The conclusion drawn from our experimentation is that LightGBM may be the most effective model in detecting credit card fraud.

More research can be done into the subject, however. Our research used data on credit card transactions from a population, rather than an individual. It is possible that Isolation Forests may be best used to detect patterns in an individual's transactional history, and detect outliers from their personal habits. LightGBM is a supervised model, so it may be forced to learn from population data. It may be the case that Isolation Forests can therefore be more tailored to an individual and perform better in this case. More research is required to be fully conclusive.

References

[1] B. Cruz, "62 million americans experienced credit card fraud last year," https://www.security.org/digital-safety/credit-card-fraud-report/, Security.org, 2025, accessed: 3 Oct. 2025.
[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.
[3] N. Fariha and Others, "Advanced fraud detection using machine learning models: Enhancing financial transaction security," *International Journal of Accounting and Economics Studies*, vol. 12, no. 2, pp. 85–104, Jun 2025.
[4] H. John and S. Naaz, "Credit card fraud detection using local outlier factor and isolation forest," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 4, pp. 1060–1064, Apr 2019.
[5] S. Ounacer and Others, "Using isolation forest in anomaly detection: The case of credit card transactions," *Periodicals of Engineering and Natural Sciences*, vol. 6, no. 2, pp. 394–400, Dec 2018. [Online]. Available: https://pen.ius.edu.ba
[6] H. Thimonier, F. Popineau, A. Rimmel, B.-L. Doan, and F. Daniel, "Comparative evaluation of anomaly detection methods for fraud detection in online credit card payments," *arXiv preprint arXiv:2312.13896*, Dec 2023. [Online]. Available: https://arxiv.org/abs/2312.13896