Credit Card Fraud Detection

Contents

I. INTRODUCTION

Anomaly detection is one of the most common tasks in Machine Learning. Like clustering, anomaly detection is also considered an unsupervised learning technique, which do not require data label, or dependent variable, to train the model. The purpose of anomaly dectection is to find out a group of noise points, or outliers, which have data patterns extremely different from the rest of the dataset. This group tends to be very small and makes the dataset imbalanced. Anomaly detection is applied in different use cases such as intrusion detection, fraud detection, health monitoring, and defect detection, (Gavrilova, 2021). For example, in banking industry, criminals tend to use credit cards as their payment method and make ambigous transactions such as huge amount in one trasaction, payment of the same amount on the same product over long time, and so on. The amount of transactions per day is many and humans can not investigate using eyes. Therefore, to solve this problem, machine learning steps in and popular algorithms used for anomaly detection can be Z-score method, Isolation Forest, DBSCAN, KMeans Clustering, One-class SVM and many more. In this paper, the dataset Credit Card Fraud Detection is used to illustrates the application of anomaly detection technique.

II. DATASET DESCRIPTION

The dataset Credit Card Fraud Detection, (Credit Card Fraud Detection, 2018), contains 284,807 transactions and only 492 transactions are flagged as fraud, which can be viewed in variable Class, 1 indicates fraud and 0 means normal. In addition, there are 28 variables, V1 to V28, which were already transformed using PCA to hide the actual data due to confidential requirements. Only two variables, Time and Amount, remain as original. The distribution of these variables can be viewed in the following table:

variable	25%	50%	75%	count	max	mean	min	std
Amount	5.6	22	77.165	284807	25691.16	88.34962	0	250.1201
Class	0	0	0	284807	1	0.001727	0	0.041527
Time	54201.5	84692	139320.5	284807	172792	94813.86	0	47488.15
V1	-0.92037	0.018109	1.315642	284807	2.45493	3.92E-15	-56.4075	1.958696
V10	-0.53543	-0.09292	0.453923	284807	23.74514	1.77E-15	-24.5883	1.08885
V11	-0.76249	-0.03276	0.739593	284807	12.01891	9.29E-16	-4.79747	1.020713
V12	-0.40557	0.140033	0.618238	284807	7.848392	-1.80E-15	-18.6837	0.999201
V13	-0.64854	-0.01357	0.662505	284807	7.126883	1.67E-15	-5.79188	0.995274
V14	-0.42557	0.050601	0.49315	284807	10.52677	1.48E-15	-19.2143	0.958596
V15	-0.58288	0.048072	0.648821	284807	8.877742	3.50E-15	-4.49894	0.915316
V16	-0.46804	0.066413	0.523296	284807	17.31511	1.39E-15	-14.1299	0.876253
V17	-0.48375	-0.06568	0.399675	284807	9.253526	-7.47E-16	-25.1628	0.849337
V18	-0.49885	-0.00364	0.500807	284807	5.041069	4.26E-16	-9.49875	0.838176
V19	-0.4563	0.003735	0.458949	284807	5.591971	9.02E-16	-7.21353	0.814041
V2	-0.59855	0.065486	0.803724	284807	22.05773	5.68E-16	-72.7157	1.651309
V20	-0.21172	-0.06248	0.133041	284807	39.4209	5.13E-16	-54.4977	0.770925
V21	-0.22839	-0.02945	0.186377	284807	27.20284	1.47E-16	-34.8304	0.734524
V22	-0.54235	0.006782	0.528554	284807	10.50309	8.04E-16	-10.9331	0.725702
V23	-0.16185	-0.01119	0.147642	284807	22.52841	5.28E-16	-44.8077	0.62446
V24	-0.35459	0.040976	0.439527	284807	4.584549	4.46E-15	-2.83663	0.605647
V25	-0.31715	0.016594	0.350716	284807	7.519589	1.43E-15	-10.2954	0.521278
V26	-0.32698	-0.05214	0.240952	284807	3.517346	1.70E-15	-2.60455	0.482227
V27	-0.07084	0.001342	0.091045	284807	31.6122	-3.66E-16	-22.5657	0.403632
V28	-0.05296	0.011244	0.07828	284807	33.84781	-1.22E-16	-15.4301	0.330083
V3	-0.89036	0.179846	1.027196	284807	9.382558	-8.76E-15	-48.3256	1.516255
V4	-0.84864	-0.01985	0.743341	284807	16.87534	2.81E-15	-5.68317	1.415869
V5	-0.6916	-0.05434	0.611926	284807	34.80167	-1.55E-15	-113.743	1.380247
V6	-0.7683	-0.27419	0.398565	284807	73.30163	2.04E-15	-26.1605	1.332271
V7	-0.55408	0.040103	0.570436	284807	120.5895	-1.70E-15	-43.5572	1.237094
V8	-0.20863	0.022358	0.327346	284807	20.00721	-1.89E-16	-73.2167	1.194353
V9	-0.6431	-0.05143	0.597139	284807	15.59499	-3.15E-15	-13.4341	1.098632

III. METHODOLOGY

To develop models, which identify the anomalous transactions, the following steps are conducted: (1) data preprocessing, (2) model development, and (3) model evaluation.

- Data preprocessing:

  The process contains three sub-steps: data standardization, feature selection, and sub-sampling. In the first sub-step, all variables are standardized into a similar scale, using the formula $z = \frac{(x-\mu)}{s}$. In the second sub-step, the Time variable is excluded by my own judgment (Time values are shown as consecutive values, which seems not promising), and then the other variables are ordered in descending based on the number of anomalies covered in 492 most distant observations by individual variables. The results are summarized in the following tables:

feature	accum_anom	accum_obs	precision	recall	f1
scaled_V12	309	492	0.628	0.628	0.628
scaled_V14	341	670	0.509	0.693	0.586916805
scaled_V17	368	806	0.457	0.748	0.567362656
scaled_V11	371	965	0.384	0.754	0.508850615
scaled_V16	376	1137	0.331	0.764	0.461888584
scaled_V18	376	1356	0.277	0.764	0.406585975
scaled_V4	380	1588	0.239	0.772	0.365000989
scaled_V3	380	1703	0.223	0.772	0.346042211
scaled_V7	381	1939	0.196	0.774	0.312791753
scaled_V10	381	2246	0.17	0.774	0.278771186
scaled_V5	381	2378	0.16	0.774	0.265182013
scaled_V1	381	2499	0.152	0.774	0.254099352
scaled_V9	381	2596	0.147	0.774	0.247074919
scaled_V8	382	2950	0.129	0.776	0.221224309
scaled_V2	382	3060	0.125	0.776	0.215316315
scaled_V19	382	3472	0.11	0.776	0.19268623
scaled_V21	384	3669	0.105	0.78	0.185084746
scaled_V22	384	3746	0.103	0.78	0.181970555
scaled_V27	384	3974	0.097	0.78	0.172542759
scaled_V25	384	4307	0.089	0.78	0.15976985
scaled_V23	385	4458	0.086	0.783	0.154978136

scaled_V15	385	4887	0.079	0.783	0.143519722
scaled_V20	385	4956	0.078	0.783	0.141867596
scaled_V24	385	5407	0.071	0.783	0.130194379
scaled_V6	385	5507	0.07	0.783	0.128511137
scaled_V26	386	5968	0.065	0.785	0.120058824
scaled_V13	386	6423	0.06	0.785	0.11147929
scaled_V28	386	6525	0.059	0.785	0.109751185
scaled_Amount	386	6688	0.058	0.785	0.10801898

The table shows that among 806 observations, which are combined among 492 most distant observations by three variables, scaled_V12, scaled_V14, and scaled_V17, there are 368 actual anomalies (a recall of 74.8%). In case, other variables are accounted for, the number of actual anomalies only increases up to 386, an increase of 4.89%, which is considered insignificant. Therefore, the three variables scaled_V12, scaled_V14, and scaled_V17 are used for the model development, while other variables are removed to reduce time-processing for model training and avoid noises. For the third sub-process, the dataset is divided into 200 samples of similar size, each sample contains transactions that occurred within consecutive timeframes (Time variable is used to cluster the dataset). These samples will be leveraged to reduce the training time for DBSCAN.

- Model development:

  Three selected algorithms to cluster the dataset are Distance-based method, Isolation Forest, and DBSCAN.

Distance-based method:

The method is a simple and naïve approach. The centroid of the whole dataset is determined by averaging the coordinates of all observations. Then, the distances between each individual observation and the centroid are calculated using Euclidean distance formula. In this approach, the anomalies tend to be the observations, which are most distant to the centroid. The strength of this approach is simple and quickly deployed, however, in some cases, most distant observations are just normal. For example, a dataset has the shape of a rugby, the top and bottom of the ball are the most distant points to the center but it is still normal. To analyze the performance of this method, different numbers of N (from 200 to 1200) most distant observations are experimented with to evaluate how it covers the actual anomalies.

Isolation Forest:

The model is developed as an ensemble method of training multiple Decision Trees on sub-samples of the dataset, (Akshara, 2023). Each Isolation Tree will grow by following the procedure. First, the tree splits the node into two branches using a random threshold value of a random feature. If the value of an observation is less than the threshold, the observation is assigned to the left branch, while other unqualified observations are assigned to the right branch. Second, the process continues till all individual observations are isolated. Third, each observation is assigned a score based on the depth of the tree to isolate that observation, a low score (or the depth is shallow) means the observation more likely to be an anomaly. In addition, the algorithm uses a contamination value to define how many percentages of anomalies are in the dataset. In this paper, different contamination values, ranging from 0.0005 to 0.0055, are experimented with to evaluate the performance of the model. The anomalies are flagged as -1. Compared to the naïve distance-based method, Isolation Forest is more time-consuming, however, it is still faster than other models such as DBSCAN, or KMeans.

DBSCAN:

The model is density-based clustering that identifies different regions of high density to form separate clusters. Such regions are determined based on the distance between core points, border points, and noise points. A core point is the point surrounded by at least $MinPts$ points within a distance of Eps, (Tan et al., 2018). $MinPts$ and Eps are defined parameters by users. Border points are the ones, which have the distance to the core point less than Eps. Noise points (or anomalies) are points, which are neither core nor border points. For anomaly detection, the group of noise points is our target and normally it is flagged as -1 in DBSCAN. In this paper, instead of training the whole dataset with DBSCAN, each sub-sample created from the preprocessing step will be fitted to DBSCAN individually. The anomalies from sub-samples are then grouped together to fit into DBSCAN once again to filter out a shortlist of anomalies. These anomalies are combined with each sub-sample to fit into DBSCAN one more time. And, anomalies, which are flagged as -1 in all 200 sub-samples are considered the final anomalies of our training. By using this method, the training time for DBSCAN can be reduced significantly. For the parameter configuration, $MinPts = 10$ is selected and different Eps values, which ranges from 0.2 to 2.2, are experimented with. The main challenges of training with DBSCAN are finding the appropriate values of $MinPts$ and Eps, and time-consuming.

- Model evaluation:

  The following metrics are used to select the appropriate number of clusters:

  - Precision: The ratio of accurate positive predictions to positive predictions. The model is good as the precision is high.
  - Recall: The ratio of accurate positive predictions to actual positive observations. The model is good as the recall is high.
  - F1-Score: The harmonic mean of precision and recall. The model is good as the F1-Score is high.
  - ROC curve: The curve is formed from multiple pairs of False Positive rate and True Positive rate. A good model has the ROC curve leaning towards to top left point (0, 1) of the plot.
  - AUC: The area under the ROC curve, which varies from 0 to 1. A good model has an AUC close to 1.

IV. RESULTS

The results of the three methods Distance-based method, Isolation Forest, and DBSCAN are summarized in the tables.

The first table recorded the performance metrics of the Distance-based method with experiments of different top N most distant observations.

Top N	Correct Anomalies	Predicted Anomalies	Actual Anomalies	Precision	Recall	F1-Score
200	168	200	492	0.840	0.341	0.486
300	253	300	492	0.843	0.514	0.639
400	301	400	492	0.753	0.612	0.675
500	347	500	492	0.694	0.705	0.700
600	374	600	492	0.623	0.760	0.685
700	391	700	492	0.559	0.795	0.656
800	401	800	492	0.501	0.815	0.621
900	407	900	492	0.452	0.827	0.585
1000	411	1000	492	0.411	0.835	0.551
1100	412	1100	492	0.375	0.837	0.518

The top N of 500 shows the best F1-Score of 0.7, which is resulted from a precision of 0.694 and a recall of 0.705.

The second table recorded the performance metrics of Isolation Forest with experiments of different contamination values.

Contamination	Correct Anomalies	Predicted Anomalies	Actual Anomalies	Precision	Recall	F1-Score
0.0005	117	142	492	0.824	0.238	0.369
0.001	240	284	492	0.845	0.488	0.619
0.0015	323	426	492	0.758	0.657	0.704
0.002	365	570	492	0.640	0.742	0.687
0.0025	398	713	492	0.558	0.809	0.661

0.003	406	853	492	0.476	0.825	0.604
0.0035	413	997	492	0.414	0.839	0.555
0.004	413	1140	492	0.362	0.839	0.506
0.0045	414	1282	492	0.323	0.841	0.467
0.005	415	1425	492	0.291	0.843	0.433

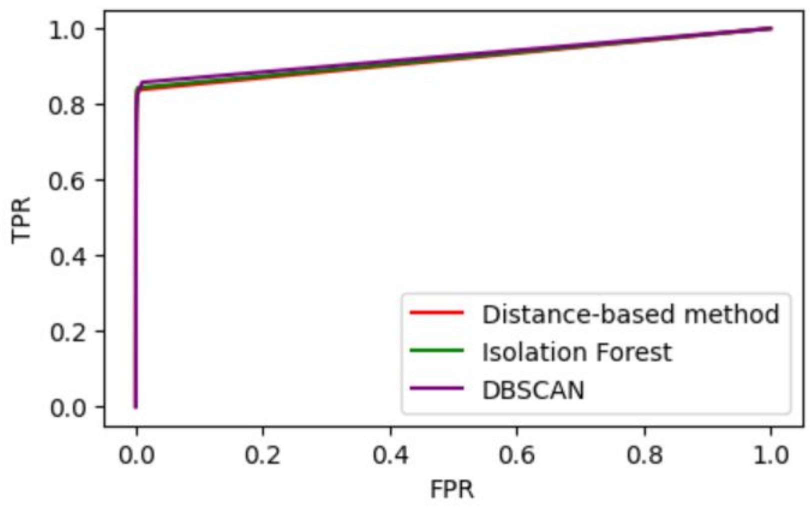
The contamination value of 0.0015 shows the best F1-Score of 0.704, which is resulted from a precision of 0.758 and a recall of 0.657.

The third table recorded the performance metrics of DBSCAN with experiments of different Eps values at $MinPts = 10$.

Eps	Correct Anomalies	Predicted Anomalies	Actual Anomalies	Precision	Recall	F1-Score
0.2	422	3274	492	0.129	0.858	0.224
0.4	405	1068	492	0.379	0.823	0.519
0.6	382	632	492	0.604	0.776	0.680
0.8	345	462	492	0.747	0.701	0.723
1	304	387	492	0.786	0.618	0.692
1.2	231	305	492	0.757	0.470	0.580
1.4	179	227	492	0.789	0.364	0.498
1.6	146	181	492	0.807	0.297	0.434
1.8	122	138	492	0.884	0.248	0.387
2	103	111	492	0.928	0.209	0.342

The Eps value of 0.8 shows the best F1-Score of 0.723, which is resulted from a precision of 0.747 and a recall of 0.701.

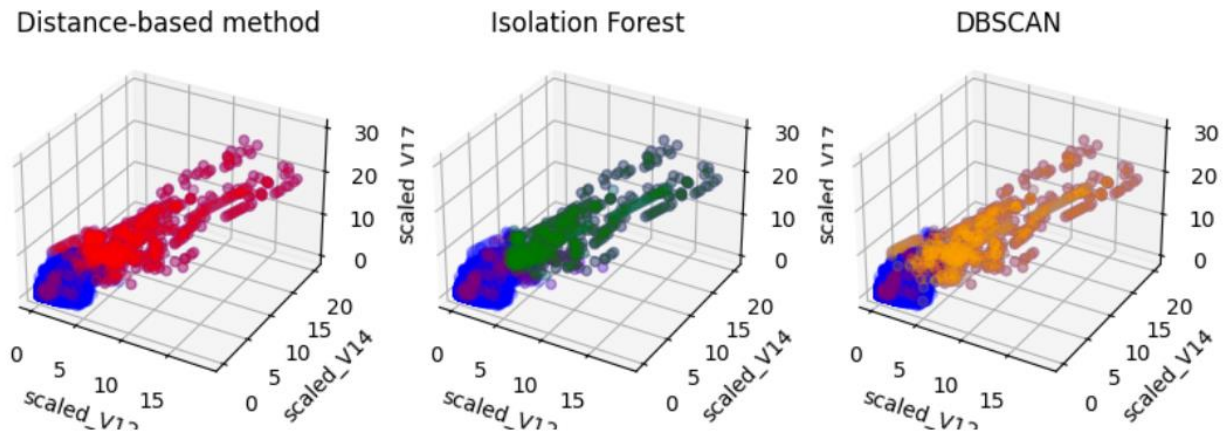
In addition, the ROC curves of the three models are presented as follows:



That also indicates the AUC of the three methods Distance-based method, Isolation Forest, and DBSCAN are 0.9182, 0.9212, and 0.9276 respectively.

V. DISCUSSION

The three methods Distance-based method, Isolation Forest, and DBSCAN seem to show similar powers in terms of identifying anomalies in credit card transactions. However, the prediction by DBSCAN is slightly better among the three models as its AUC (0.9276) and F1-Score (0.723) are the highest.



Looking at the above graph, the predictions of the three models concentrate on observations, which are far from the center of the majority area. While, most of unidentified anomalies (in purple color) gather at the center, which is likely to be ignored. Such characteristic possibly suggests for future study that other features than V12, V14, and V17 should be further investigated to see whether those unidentified anomalies are discovered in any other dimensions.

In terms of training time by individual models, the distance-based method was the fastest one, which takes under 1 second to generate the prediction. While Isolation Forest takes around 23 seconds on average for a single training. And, DBSCAN with the leverage of batch training, the average training time is around 26 seconds, which is quite close to Isolation Forest. The limitation on training time is the main challenge of Isolation Forest and DBSCAN in comparison with distance-based method. However, in other shape of dataset, in which the anomalies are not observations being most distant, other models can outperform distance-based method.

VI. CONCLUSION

The three anomaly dectection techniques, which include Distance-based method, Isolation Forest, and DBSCAN are demonstrated through using the Credit Card Fraud Dectection dataset. The three methods tend to predict observations, which are distant from the center of the dataset, to be anomalies. Among the three model, DBSCAN shows better performance with precision at 0.747 and recall at 0.701. However, the limitation of DBSCAN is time-consuming, and batch training will help improve training speed. For anomaly detection, other than the challenge of accurate prediction, the consideration into dealing with huge amount of transactions should also be taken seriously in order to generate the prediction on a timely manner, which is a requirement in actual applications in real life.

References

Gavrilova, Y. (2021). What is anomaly detection in machine learning? *Serokell Software Development Company*. https://serokell.io/blog/anomaly-detection-in-machine-learning

Credit card fraud Detection. (2018, March 23). Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Tan, P., Steinbach, M., Karpatne, A., & Kumar, V. (2018). Introduction to Data Mining (2nd Edition). *Introduction to Data Mining*. https://dl.acm.org/citation.cfm?id=3208440

Akshara. (2023). Anomaly detection using Isolation Forest – A Complete Guide. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/07/anomaly-detection-using-isolation-forest-a-complete-guide/