Analysis of the Influence of Parameters on Anomaly Detection Models of PyCaret

Shwetha Prabhu[1] and Renuka A[2]

[1]*Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India 576104.*
[2]*Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka, India 576104.*

Abstract

Anomaly detection plays a vital role in domains such as fraud prevention, medical diagnostics, and network security by identifying data points that deviate from normal patterns. PyCaret, a Python-based machine learning library, offers multiple unsupervised models for anomaly detection. This study evaluates the performance of five PyCaret models—K-Nearest Neighbours, Isolation Forest, Local Outlier Factor, Cluster-Based Local Outlier Factor, and Histogram-Based Outlier Detection—on a publicly available financial transaction dataset. Among these, the Isolation Forest model performed best and was further analysed for the effect of its *contamination* parameter. Varying the contamination parameter from 0.01 to 0.1 revealed that model performance improves when this threshold is set based on the statistical distribution of outliers in the dataset.

Keywords: Anomaly detection, unsupervised learning models, Isolation Forest, PyCaret, contamination parameter.

River Journal, 43–51.

1 Introduction

Anomaly detection refers to identifying data points that deviate significantly from the majority—often called outliers [1]. In many traditional classification tasks, such data are filtered out as noise, whereas anomaly detection specifically focuses on uncovering such rare and informative events. Its relevance spans diverse domains including fraud detection, network intrusion, medical diagnostics, and industrial quality control. For example, anomaly detection helps flag suspicious transactions in financial systems, detect abnormal ECG or EEG patterns in healthcare [2], and identify manufacturing defects. Machine learning techniques for anomaly detection are broadly categorized into supervised, semi-supervised, and unsupervised methods [1]. Supervised models require labeled data and perform well when anomalies are well-defined. However, in real-world applications, such labels are often unavailable. Semi-supervised models are trained on normal data and flag deviations during inference. Unsupervised models, requiring no labels, analyse the structure of the data and assign anomaly scores to each instance, making them ideal for unknown or evolving scenarios.

PyCaret [3] is a low-code Python library that simplifies machine learning workflows and supports several unsupervised anomaly detection algorithms. In this study, five commonly used PyCaret models are evaluated and compared: Isolation Forest, K-Nearest Neighbours, Local Outlier Factor, Cluster-Based Local Outlier Factor, and Histogram-Based Outlier Score. Isolation Forest (iForest) [4] isolates anomalies through recursive data partitioning. Anomalous instances, which are easier to isolate, result in shorter paths in the isolation tree. K-Nearest Neighbours (KNN) [6], although typically supervised, can be adapted for unsupervised anomaly detection by computing distances to neighbouring points—larger distances often indicating anomalies. Local Outlier Factor (LOF) [5] compares the local density of each point with its neighbours; low-density points receive high LOF scores and are treated as outliers. Cluster-Based LOF (CBLOF) [7] involves clustering the data and assigning anomaly scores based on intra- and inter-cluster distances, distinguishing between small and large clusters. Histogram-Based Outlier Score (HBOS) [8] models the distribution of each feature using histograms and assumes feature independence. It identifies outliers based on unusually low-frequency combinations. This study compares these models on publicly available fraud datasets and further investigates the effect of the contamination parameter on the Isolation Forest model.

As this parameter controls the threshold that separates normal data from anomalies, its tuning is essential for reliable detection.

2 Literature Review

Anomaly detection has been widely explored using supervised, semi-supervised, and unsupervised machine learning approaches. Belavagi and Muniyal [9] evaluated the performance of various supervised learning algorithms for intrusion detection and found that Random Forest achieved the highest accuracy. Alsulaiman and Al-Ahmadi [10] compared supervised and unsupervised methods, concluding that supervised models generally yield better performance in terms of accuracy. Krsteski et al. [11] investigated deep learning models such as CNN, RNN, DNN, and CNN–RNN hybrids using the CICIDS 2017 dataset and reported that CNN achieved the highest anomaly detection accuracy of 98.75%. Salim and Oughdir [12] evaluated deep learning techniques for detecting DoS attacks in wireless sensor networks, highlighting the superior capabilities of such models in complex detection scenarios. Hilal et al. [13] presented a review of anomaly detection techniques for financial fraud and discussed modern trends and challenges in the field. Pourhabibi et al. [14] focused on graph-based anomaly detection, conducting a systematic review using metrics such as processing time, precision, recall, accuracy, and F1-score. Quincozes et al. [15] demonstrated that supervised models outperformed unsupervised ones for blackhole attack detection, with the REPTree classifier achieving the highest F1-score. Lee et al. [16] applied the Isolation Forest algorithm with hyperparameter tuning to detect anomalies in storage batteries, showing that the method was both efficient and scalable. Olaniyan and Owoseni [17] conducted a comparative analysis of PyCaret's unsupervised anomaly detection models and the WHO's Data Quality Review (DQR) toolkit using HIV/AIDS datasets from Africa. Their study suggests that integrating anomaly detection tools with DQR can significantly enhance Data Quality Assessment (DQA) processes.

3 Methodology

As discussed in the literature, supervised learning algorithms tend to perform well in terms of accuracy. However, anomalies are often unknown and previously unseen in real-world applications such as fraud detection.

In such cases, supervised models alone may not be sufficient, and there arises a need to adopt unsupervised anomaly detection techniques. However, the parameters used significantly affect the performance of these unsupervised models. This study focuses on analyzing the influence of the contamination parameter, which plays a key role in determining the threshold for classifying a data point as normal or anomalous. PyCaret is chosen for implementation because it is an open-source, Python-based machine-learning library that simplifies the experimentation process. It provides access to several unsupervised anomaly detection models and allows users to perform end-to-end modeling with minimal coding. Although anomaly detection has several real-life applications, this work focuses specifically on fraud detection due to its importance in mitigating financial losses. The objective of the study is to analyze the behavior of anomaly detection models when their parameters are varied. The analysis is not tied to a specific dataset, and two publicly available datasets have been used for experimentation: a financial transactions dataset and a credit card transactions dataset, both sourced from Kaggle. A subset of 15,000 samples was taken from the financial transactions dataset, which includes fields such as transaction amount, transaction time, transaction ID, terminal ID, and a fraud indicator. The credit card transactions dataset contains one million samples and consists of features such as distance from home, distance from the last transaction, ratio to median purchase, repeat retailer, used chip, used pin number, and online order, along with a label indicating whether the transaction was fraudulent. The first three features are categorical, and the rest are binary. All the datasets used in this study are imbalanced, with a significantly lower number of fraudulent samples than normal ones. PyCaret automatically handles missing values during the setup phase, and no missing values were found in the datasets used. It also provides an option to ignore features that are not relevant to the modeling process. In this work, the fraud label was excluded during training as it is not required for unsupervised models. The anomaly detection module in PyCaret is designed to identify rare or suspicious events that significantly deviate from normal patterns. It includes built-in preprocessing capabilities that are applied during the setup. The modeling process includes setting up the environment, training the model, assigning anomaly labels, analyzing model performance, predicting anomalies on new data, and saving the final model. Each data point receives an anomaly score and a binary label, where ‘1’ indicates an anomaly and ‘0’ indicates a normal data point. These outputs are used to evaluate the effect of varying the contamination parameter on model performance.

Table 1 Comparison of various unsupervised models.

Model	Anomalous (Train)	Anomalous (Test)	Accuracy (%)
Isolation Forest	85,500	4,473	94.2
K-Nearest Neighbours	85,500	4,583	91.6
Local Outlier Factor	85,500	5,384	72.6
Cluster-Based Local Outlier	85,500	4,359	96.9
Histogram-Based Local Outlier	72,497	3,797	89.8

4 Results and Discussion

The anomaly detection models implemented using PyCaret include Isolation Forest, K-Nearest Neighbours, Local Outlier Factor, Cluster-Based Local Outlier Factor, and Histogram-Based Local Outlier Detection. All these models are based on unsupervised learning. In this study, 95% of the data was used for training and 5% was reserved for testing. Though these models do not rely on labeled data, the split was done to evaluate their performance on unseen data. All models were tested with the contamination parameter fixed at 0.09, and the rest of the parameters were left at default values. The credit-card transaction dataset consisted of 1,000,000 records, out of which 950,000 samples were used for training and 50,000 for testing. Among the test samples, 4,228 were anomalous, and the training set contained 83,115 anomalies. The number of anomalous samples identified and the corresponding accuracy for each model are presented in Table 1. From Table 1, it is observed that the Cluster-Based Local Outlier model provided the best accuracy at 96.9%, followed closely by Isolation Forest at 94.2%. However, the performance of the cluster-based model is sensitive to the number of clusters (k), and incorrect settings may degrade accuracy. A similar dependency on k exists in the K-Nearest Neighbours and Local Outlier Factor models. In contrast, the Histogram-Based model was less effective when the number of anomalies was small. Owing to its robustness and popularity in the literature, the Isolation Forest model was chosen for further analysis. In the second phase of the experiments, the behaviour of Isolation Forest was studied by varying the contamination parameter from 0.01 to 0.1 while keeping other parameters at default values [4]. This parameter determines the fraction of data expected to be anomalous and is used internally to compute a threshold based on the anomaly scores. For example, a contamination value of 0.05 implies that 5% of the data are assumed to be outliers. The fraudulent transaction dataset containing 15,000 records was used, with 5% of the data allocated for testing. The following performance metrics were computed:

Table 2 Accuracy for varying contamination parameter values of Isolation Forest.

Contamination	Accuracy (Train)	Accuracy (Test)
0.01	90.07	88.93
0.03	89.18	89.18
0.05	86.69	87.87
0.07	91.13	88.90
0.09	95.50	95.40

- **True Positive (TP)**: actual fraud correctly predicted as fraud.
- **True Negative (TN)**: normal data correctly predicted as normal.
- **False Positive (FP)**: normal data incorrectly predicted as fraud.
- **False Negative (FN)**: fraud data incorrectly predicted as normal.

The standard performance metrics used in evaluating model effectiveness include accuracy, precision, recall, and the F1 score. Accuracy, given in Eq. (1), measures the proportion of correctly classified instances out of all instances. Precision (Eq. (2)) indicates the fraction of correctly identified frauds among all predicted frauds. Recall, as expressed in Eq. (3), measures the ability of the model to detect actual fraudulent transactions. The F1 Score, shown in Eq. (4), is the harmonic mean of precision and recall, providing a balanced metric that is particularly useful for imbalanced datasets.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Although accuracy provides a general overview of performance, it is not suitable for imbalanced datasets. Hence, metrics like precision, recall, and F1 score were also considered. The accuracy for various contamination values is shown in Table 2. Detailed performance metrics for the Isolation Forest model on test and training data are provided in Tables 3 and 4. As observed from the tables, the accuracy and F1 score are highest when the contamination parameter is set to 0.09.

Table 3 Performance metrics for Isolation Forest on test data.

Cont.	TP	TN	FP	FN	Recall	Precision	F1 Score	Accuracy
0.01	537	11948	888	878	37.95	37.60	37.5	88.93
0.03	331	12379	97	1444	16.70	77.30	27.8	89.18
0.05	499	12281	713	1254	28.40	41.10	33.5	87.87
0.07	805	12296	193	957	45.70	80.60	60.2	88.90
0.09	1207	12413	72	559	68.30	94.30	79.2	95.40

Table 4 Performance metrics for Isolation Forest on training data.

Cont.	TP	TN	FP	FN	Recall	Precision	F1 Score	Accuracy
0.01	43	624	26	57	43.00	62.30	50.7	90.07
0.03	13	656	3	78	14.30	81.25	23.8	89.18
0.05	30	629	14	77	28.03	68.10	39.6	86.69
0.07	33	634	12	71	31.70	73.33	43.5	91.13
0.09	69	647	3	31	69.00	95.83	80.0	95.50

The dataset used contains approximately 12% anomalies, which aligns closely with the best-performing contamination value. This suggests that the optimal setting of the contamination parameter can significantly improve the detection capability of the model, especially when aligned with the actual proportion of outliers in the dataset.

5 Conclusion

This study presented a comparative analysis of unsupervised anomaly detection models available in PyCaret, with a specific focus on their performance in fraud detection scenarios. Among the models evaluated, the Isolation Forest model exhibited consistently high accuracy and was further explored to assess the effect of varying the contamination parameter. The results demonstrated that the model performs optimally when the contamination parameter is aligned with the statistical distribution of outliers in the dataset. This highlights the importance of tuning the contamination parameter based on the expected proportion of anomalies, particularly in imbalanced datasets. For real-world applications where dataset characteristics may not be explicitly known, prior statistical knowledge or estimation techniques could guide parameter selection. As part of future work, additional parameters of the Isolation Forest model will be examined, and experiments will be extended to larger and more diverse datasets. Given the wide-ranging applications of anomaly detection, future studies will also consider domain-specific datasets to evaluate model behaviour across different contexts.

References

[1] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLOS ONE, 11(4), 2016.*

[2] M. Radhakrishnan, S. Boruah, and K. Ramamurthy. EEG-based anomaly detection for autistic kids – a pilot study. Traitement du Signal, 39(3):1005–1012, 2022.

[3] M. Ali. PyCaret: An open source, low-code machine learning library in Python, 2020.

[4] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In *Proceedings of the IEEE International Conference on Data Mining, Beijing, China, 2008.*

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 93–104, 2000.

[6] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer. KNN model-based approach in classification. In *Lecture Notes in Computer Science, vol. 2888. Springer, Berlin, Heidelberg, 2003.*

[7] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. Pattern Recognition Letters, 24(9–10):1641–1650, 2003.

[8] M. Goldstein and A. Dengel. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In *KI 2012, Poster and Demo Track, pages 59–63, 2012.*

[9] C. M. Belavagi and B. Muniyal. Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89:117–123, 2016.

[10] L. Alsulaiman and S. Al-Ahmadi. Performance evaluation of machine learning techniques for DoS detection in wireless sensor network. *International Journal of Network Security & Its Applications, 13(2):21–29, 2021.*

[11] S. Krsteski, M. Tashkovska, B. Sazdov, L. Radojichikj, A. Cholakoska, and D. Efnusheva. Intrusion detection with supervised and unsupervised learning using PyCaret over CICIDS 2017 dataset. In Lecture Notes in Networks and Systems, vol. 724. Springer, Cham, 2023.

[12] S. Salim and L. Oughdir. Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network. *Journal of Big Data, 10(17), 2023.*

[13] W. Hilal, S. A. Gadsden, and J. Yawney. Financial fraud: A review of anomaly detection techniques and recent advances. Expert Systems with Applications, 193, 2022.

[14] T. Pourhabibi, K.-L. Ong, B. H. Kam, and Y. L. Boo. Fraud detection: A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems, 133, 2020.*

[15] S. E. Quincozes, J. F. Kazienko, and V. E. Quincozes. An extended evaluation on machine learning techniques for denial-of-service detection in wireless sensor networks. Internet of Things, 22, 2023.

[16] C.-H. Lee, X. Lu, X. Lin, H. Tao, Y. Xue, and C. Wu. Anomaly detection of storage battery based on isolation forest and hyperparameter tuning. In *Proceedings of the International Conference on Mathematics and Artificial Intelligence, ACM, New York, pages 229–233, 2020.*

[17] F. M. A. Olaniyan and A. Owoseni. Toward improved data quality in public health: Analysis of anomaly detection tools applied to HIV/AIDS data in Africa. In Proceedings of the IST-Africa Conference, Ireland, pages 1–9, 2022.

[18] S. Singh. Fraudulent transaction detection. Available: https://www.kaggle.com/datasets/sanskar457/fraud-transaction-detection, Accessed: Jul. 12, 2023.
[19] D. Narayanan R. Credit card fraud analysis. Available: https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud, Accessed: Jul. 12, 2023.

Biography

Shwetha Prabhu received the B.Tech. degree in Electronics and Communication Engineering from Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She has presented papers at international conferences. Her research interests include medical image processing, cryptography, and information security.

Dr. Renuka A. received the B.E. and M.Tech. degrees from the University of Mysore, and the Ph.D. degree from NITK Surathkal, India. She is currently a Professor in the Department of Computer Science and Engineering at Manipal Institute of Technology, MAHE. She has published several papers in reputed international journals and conferences. Her research interests include networks, cybersecurity, and computer vision. She also serves as a reviewer for several international journals.