Anomaly Detection Algorithms: Comparative Analysis and Explainability Perspectives

Sadeq Darrab[1](✉), Harshitha Allipilli[1], Sana Ghani[1], Harikrishnan Changaramkulath[1], Sricharan Koneru[1], David Broneske[2], and Gunter Saake[1]

[1] University of Magdeburg, Magdeburg, Germany
sadeq.darrab@ovgu.de,
{harshitha.allipilli,harikrishnan.changaramkulath}@st.ovgu.de,
{sana.ghani,sricharan.koneru}@ovgu.de, gunter.saake@ovgu.de
[2] German Centre for Higher Education Research and Science Studies, Hanover, Germany
broneske@dzhw.eu

Abstract. In order to detect outliers and potential anomalies in datasets, anomaly detection plays a pivotal role in identifying infrequent and irregular occurrences. The purpose of this paper is to examine and compare the effectiveness of prominent anomaly detection algorithms, including Isolation Forest, Local Outlier Factor (LOF), and One-Class Support Vector Machines (SVM). A variety of datasets are used in our assessment to evaluate key metrics such as precision, recall, F1-score, and overall accuracy. We also introduce innovative techniques that enhance the interpretability of these algorithms, shedding light on the underlying factors that contribute to anomaly detection. By providing insights into the attributes and behaviors associated with anomalies, our research empowers decision-makers to cultivate a profound comprehension of the identified anomalies, subsequently facilitating well-informed decisions grounded in the outcomes of anomaly detection. Through our meticulous comparative analysis and our dedication to unraveling the elements of explainability, we provide invaluable perspectives and pragmatic suggestions to facilitate effective anomaly detection in real-world scenarios.

Keywords: Anomaly Detection · Unsupervised Learning · Explainability

1 Introduction

Anomalies are outliers, noise, exceptions, and deviations from the real behavior of the system. Detecting anomalies involves identifying objects, patterns, occurrences, and observations that do not follow an anticipated pattern [6]. Anomaly detection plays a crucial role in various domains, including Cybersecurity, fraud detection, industrial monitoring, and healthcare. It is possible to identify outliers and potential anomalies that require special attention by identifying and

D. Benavides-Prado et al. (Eds.): AusDM 2023, CCIS 1943, pp. 90–104, 2024.
https://doi.org/10.1007/978-981-99-8696-5_7

flagging rare and irregular instances within datasets. The traditional supervised learning approach relies on labeled data, making it less suitable for detecting anomalies in unsupervised datasets. Due to their ability to uncover unknown anomalies without relying on pre-labeled data, unsupervised anomaly detection algorithms have gained increasing attention [15].

Anomaly detection methods have been found to be useful in a variety of fields, but they are not without their challenges. Since anomalies are often rare and difficult to identify in real-world scenarios, obtaining labeled data for anomalies can be challenging [12]. Furthermore, conventional anomaly detection methods may face difficulties in explaining the anomalies detected in the data. Many existing anomaly detection methods, often operate as "black-box" models, providing little to no insight into how they arrive at their anomaly detection decisions. This lack of transparency and interpretability hinders the adoption of these algorithms in critical domains where explanations for detected anomalies are essential for decision-making [4].

In our paper, we investigate unsupervised anomaly detection algorithms and explore their effectiveness in identifying anomalies within diverse datasets, as well as their limitations. In this paper, we focus on three widely used algorithms: isolation forest, local outlier factor (LOF), and one-class support vector machines (SVM). In terms of computational efficiency, scalability, and interpretability, these algorithms leverage different techniques to detect anomalies.

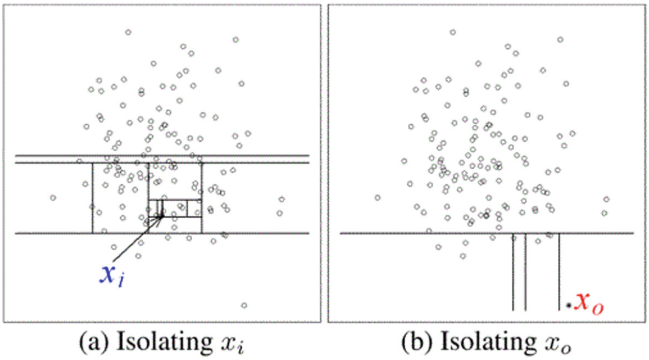
The main contribution of this paper can be summarized as follows.

- We provide a comprehensive analysis and comparison of these algorithms, evaluating their performance metrics such as precision, recall, F1-score, and overall accuracy on various datasets.
- Our study is focused on benchmarking a variety of algorithms against each other within the realm of anomaly detection. The primary objective is to discern and highlight the unique strengths and weaknesses exhibited by these algorithms across diverse anomaly detection scenarios.
- Through an exploration of novel techniques, we explore how to make algorithms more interpretable. By uncovering the underlying factors contributing to detected anomalies, we empower decision-makers to gain deeper insights into the anomalies detected and make informed decisions based on the anomaly detection results.
- We bridge the gap between algorithmic efficiency and human interpretability by combining performance evaluation with explainability.

This paper has the following structure. The background of anomaly detection models is discussed in Sect. 2. The methodology used in this paper is described in Sect. 3. The detailed implementation of data preprocessing, model training and evaluation, and explainability is presented in Sect. 4. In Sect. 5, we present the evaluation results of these models and datasets whereas in Sect. 6, we present a review of the exiting work. Our final section concludes our paper and outlines our future plans.

2 Background

Data mining and machine learning use anomaly detection to identify occurrences that are significantly different from the norm. Since machine learning has gained popularity in anomaly identification, both supervised and unsupervised learning methods have been used to capture complicated patterns and identify anomalies. For tasks, where anomalies are identified in the training data, supervised learning techniques such as Naive Bayes, Support Vector Machines (SVM) and K-Nearest Neighbors have been investigated and compared again, and One Class SVM, an unsupervised anomaly detection algorithm that outperforms these supervised algorithms [7]. These algorithms may face difficulties with labeled data availability in real-world applications where anomalies are uncommon and difficult to obtain. Hence, we focus on anomaly detection methods based on unsupervised learning techniques that do not require labeled training data. Unsupervised methods such as Local Outlier Factor (LOF), Isolation Forest, and One-Class SVM are used.



(a) Isolating x_i (b) Isolating x_o

Fig. 1. Isolation Forest [9]

2.1 Isolation Forest

Isolation Forest is a well-known ensemble-based anomaly detection algorithm lauded for its efficiency and capacity to handle high-dimensional data. This algorithm operates by isolating anomalies into partitions in a random forest-like manner. Decision trees are constructed by recursively selecting random features and splitting randomly, which isolates the anomalies effectively.

In Fig. 1(a), a relatively small number of anomalies is visible. This results in the formation of smaller partitions and shorter paths within the tree structure. Contrarily, Fig. 1(b) demonstrates instances characterized by distinct attribute values, leading to early partitioning. Notably, anomalies are likely to be swiftly isolated within a few steps, thereby setting them apart from the larger cluster of data points, which necessitate more partitions for segregation. The path length traversed by an instance within the tree serves as an anomaly score, where a shorter path corresponds to higher anomaly levels. This scoring mechanism enables efficient differentiation between anomalies and normal data points.

2.2 Local Outlier Factor (LOF)

LOF is a density-based anomaly detection algorithm that assesses the local density of instances relative to their neighbors. It calculates the local reachability density for each data point by comparing its distance to its k-nearest neighbors. The LOF score indicates how much an instance's density deviates from that of its neighbors. Low LOF scores correspond to points with significantly lower densities than their neighbors, indicating anomalies [3]. Figure 2 illustrates the LOF mechanism: for data point o2, its local density is computed using its k-nearest neighbors; if o2 has lower density than its neighbors, it is labeled as an anomaly.

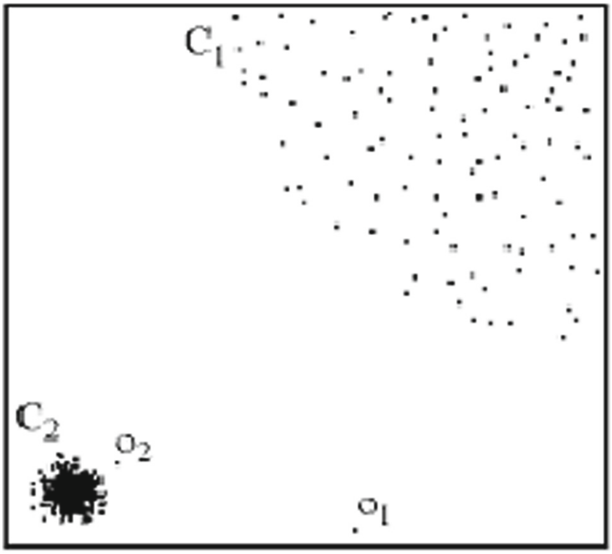


Fig. 2. Local Outlier Factor [1]

2.3 One-Class Support Vector Machine (One-Class SVM)

One-Class SVM is a widely-used algorithm for unsupervised anomaly detection. It is based on the principles of Support Vector Data Description (SVDD). Its primary objective is to learn a hyper-sphere to characterize a single class of data points. Instances outside the decision boundary are classified as anomalies [2]. In Fig. 3, the working principle of a One Class Support Vector Machine is depicted.

Points near the origin are classified as anomalies (-1), while all other points are considered normal.

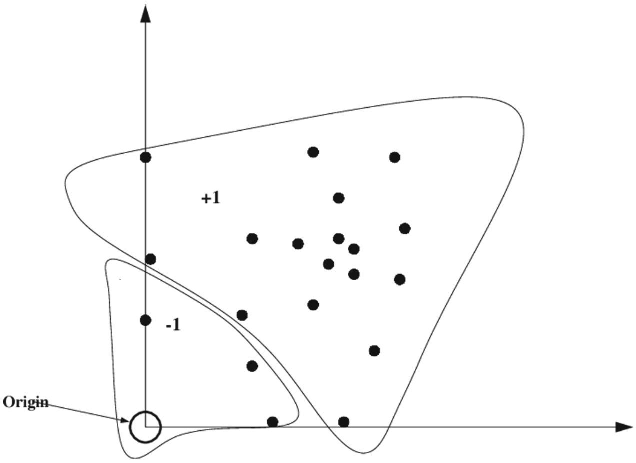


Fig. 3. One Class Support Vector Machine [10]

This paper evaluates model performance based on precision, recall, F1-score, and accuracy metrics. Accuracy reflects the correctness of normal and anomaly classifications, while precision gauges accurate positive predictions, emphasizing correct identification without false positives. Recall measures the model's ability to identify actual positives within the dataset, vital for imbalanced scenarios, and F1-score balances precision and recall, proving valuable for uneven data. Given the rarity of anomalies and their significant deviations, a focus on recall is crucial to minimize false negatives, ensuring robust real-world anomaly detection. The use of F1-score addresses imbalanced data's challenges, accounting for false positives and negatives. The implementation leverages Anchors, an interpretable rule extraction method, chosen for its simplicity and high precision, enhancing the model's trustworthiness.

The research encompasses cardiovascular disease and credit card fraud datasets, addressing class imbalances and distinct data distributions. Comparative analysis sheds light on algorithm effectiveness, offering insights into various contextual advantages and disadvantages. Additionally, the paper highlights explainability's pivotal role in anomaly detection, striving to demystify algorithm decisions through interpretable methodologies. By developing unsupervised anomaly detection algorithms and providing practical guidance, the research empowers users to select optimal approaches tailored to their unique scenarios.

3 Methodology

The main purpose of this research paper is to identify anomalies using different models and explain why these models made this decision. Figure 4 shows a the workflow of the proposed model. This method consists of the following steps.

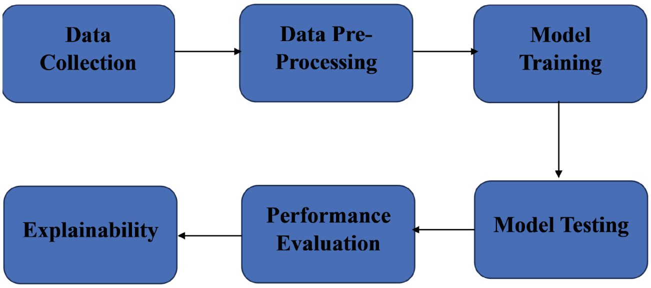


Fig. 4. Graphical Representation of Proposed Methodology

- **Data Collection:** To begin with, we have carefully selected datasets that contain anomalies, including Heart Disease Prediction and Credit Card Fraud detection. These datasets are both expansive and diverse, providing a comprehensive testing ground for the subsequent steps.
- **Data Pre-processing:** The second step involves data pre-processing. A judicious selection of pertinent features becomes imperative for optimizing model performance when recognizing the presence of both useful and extraneous features in the dataset. Due to the heterogeneous nature of the data, consisting of both categorical and continuous attributes, adept handling is required. In order to overcome this, a two-pronged approach is adopted: discretization of continuous values to impart discreteness, followed by feature encoding, such as one-hot encoding and label encoding, for categorical attributes. In order to ensure homogeneity and comparability, normalization or standardization techniques are used.
- **Model Training:** This step involves training the model. As outlined in Sect. 2, the three models are trained here for practical implementation. To pinpoint the optimal hyperparameters for each dataset-model pair, meticulous experiments are conducted.
- **Model Testing:** Testing the model takes place in the fourth step. In order to assess the models, a distinct test dataset is used, which was not used during training. As part of the performance evaluation, the models are scrutinized against the defined evaluation metrics - precision, recall, F1-score, and accuracy.
- **Performance Evaluation:** In this step, the efficacy of the model is comprehensively evaluated. Leveraging the outcomes of the previous phase, the models are subjected to rigorous evaluation across different test datasets. Performance metrics, including precision, recall, F1-score, and accuracy, form the

bedrock of assessment, interlacing both the individual model strengths and trade-offs. Based on these evaluations, along with training times and recall scores, informed comparisons can be made.
- **Interpretability:** In the final phase of this research journey, we will examine the interpretability of model outputs. After training the models, we utilize rule-generating methodologies, such as Anchors, to generate intelligible and human-readable decision rules, such as if-then statements. Our ability to understand the rationale behind the models' classifications is enhanced by this transparency.

For our research to be successful, it is imperative that these interlocking steps are meticulously orchestrated. In the subsequent Implementation and Evaluation section, each phase will be meticulously analyzed, encompassing the intricate details that collectively propel us toward our goals.

4 Implementation and Evaluation

This section outlines the implementation of the mentioned unsupervised machine learning algorithms for anomaly detection on diverse datasets: Heart Disease and Credit Card Fraud from Kaggle. We conduct a comparative analysis to assess Isolation Forest, Local Outlier Factor, and One-class SVM's anomaly detection performance and generalization across real-world scenarios. It encompasses data pre-processing, hyper-parameter tuning, model evaluation, and comparative effectiveness analysis with other algorithms.

4.1 Datasets

The Heart Disease dataset [16] contains 70000 instances and 11 features. The second dataset [5], the Credit Card Fraud, contains 1296675 instances and 22 features. We balanced the dataset by sampling 20,000 instances from the majority class (normal) and 5,000 instances from the minority class (fraud).

4.2 Data Pre-processing

Several data preprocessing steps were performed in this section to ensure the data's quality and suitability for modeling:

- Data Cleaning: In this stage, we focused on identifying and resolving errors, inconsistencies, and missing values present in the dataset. We took necessary steps to handle duplicate values, null values, and inconsistent data entries. Fortunately, there were no null values in the provided datasets. However, we did encounter an inconsistency in the heart disease dataset, specifically in the features *ap_hi* and *ap_lo*, (since systolic and diastolic blood pressure cannot be negative) where some values were negative. We promptly addressed and corrected such discrepancies to ensure data integrity.

For the heart disease dataset, we converted the age feature from days to years for better interpretation and the height feature from centimeters to feet for standardization as part of the data pre-processing step. Additionally, we dropped the 'id' column as it does not contribute to the anomaly detection task.

- Data Transformation: We used the following techniques for data transformation:
  - Log Transformation: To address data skewness and reduce the impact of extreme values, we employed the natural logarithm of the data values. Specifically, we applied this technique to the *amt* and *city_pop* features in the Credit Card Fraud Dataset. Prior to the transformation, the *amt* feature had minimum and maximum values of 1 and 28,949, respectively, while the *city_pop* feature ranged from 23 to 2,906,700. After applying the Log Transformation, the minimum and maximum values of *amt* were adjusted to 0 and 10, respectively, and the *city_pop* feature ranged from 3 to 15. This transformation helped to normalize the data and mitigate the effects of extreme values, thereby improving the analysis.
  - Scaling: To bring all features to a common range and standardize the data, we utilized min-max scaling, also known as normalization. Specifically, we applied this operation to four features in the Credit Card Fraud Dataset: *lat*, *long*, *merch_lat*, and *merch_long*. Before scaling, the mean values for *lat* and *merch_lat* were 39, while the means for *long* and *merch_long* were -90. After applying the min-max scaler, the means of these features were adjusted to 0, effectively standardizing the data and ensuring that they fall within the same range. This normalization process facilitates more consistent and reliable analyses across the dataset.
  - Encoding Categorical Variables: To facilitate effective interpretation by the models, we converted categorical variables into numerical representation using label encoding. Within the credit card fraud dataset, we identified 14 unique categories of transactions, 2 genders, and 50 states, all represented with strings. Through label encoding, we transformed the features *category*, *gender* and *state* into corresponding integers, enabling the models to handle them efficiently during training and prediction processes.
- Feature Engineering: As part of this crucial step, we focused on extracting relevant information from the raw data and creating new features to improve the model's learning capabilities. Our objective was to provide more meaningful and informative input for the models to better understand and process the data. For instance, in the heart disease dataset, we converted the *age* feature from days to years for better interpretability. Additionally, we transformed the *height* feature from centimeters to feet to ensure consistency and ease of understanding. Similarly, in the credit card fraud dataset, we performed feature engineering on the *trans_date_trans_time* feature by splitting it into separate components such as hour, day, and month. This transformation enabled the models to capture temporal patterns more effectively and gain deeper insights into the data's temporal dynamics.

4.3 Algorithm Implementations

- Isolation Forest: The python implementation of isolation forest from sci-kit learn was utilized in our experiments. Among the nine available parameters, we focused on tuning two specific attributes: *n_estimators* and *contamination.*

  The parameter *n_estimators* determines the number of estimators constructed in the ensemble. After conducting thorough experiments, we observed that the default value of 100 for *n_estimators* was highly efficient for our datasets. As a result, we decided to retain the default value for this parameter, ensuring that the algorithm maintains its computational efficiency. The *contamination* parameter, on the other hand, represents the proportion of outliers present in the dataset. To ensure accurate anomaly detection, we meticulously adjusted the *contamination* parameter to match the specific characteristics of each dataset. By carefully setting this parameter, we aimed to strike a balance between detecting true anomalies while minimizing false positives.

  To assess the effectiveness of the Isolation Forest model, we generated outlier scores for instances using the *decision function* method. Subsequently, we defined a threshold of 0 to classify instances as anomalies or normal data. Based on the outlier scores, we converted the predictions into binary labels, where 1 represents anomalies and 0 denotes normal instances.

  In our experiments, we found that isolation forest is highly efficient and adept at handling large, high-dimensional datasets, requiring less memory due to its storage of random partition structures. This makes it a practical option for anomaly detection, especially with extensive datasets and high-dimensional data.
- One Class Support Vector Machine: In our study, we utilized the implementation of One Class SVM provided by the sci-kit learn library in Python. The implementation offers ten parameters that can be configured and optimized. Among these parameters, we kept the default settings for eight, and focused on adjusting two key parameters.

  The first parameter we modified was the *kernel* parameter, which allowed us to choose from four available kernels: *linear*, *poly*, *rbf*, and *sigmoid.* Through experimentation, we found that the Radial Bias Function (RBF) kernel yielded the most promising results for both datasets. Kernels play a crucial role by transforming the feature space, enabling the data to become linearly separable, and consequently improving the SVM's classification performance [14].

  The second parameter we adjusted was *nu* which represents the fraction of training errors or, in other words, the number of anomalous instances within the margin. By tuning this parameter, we could control the trade-off between maximizing the margin and capturing the anomalies effectively.

  One noteworthy observation was that as training instances increased, the One Class SVM's training time grew exponentially due to the extensive calculations required for the distance matrix. This highlights the importance of considering computational cost for efficient training with larger datasets.

- Local Outlier Factor: Local Outlier Factor (LOF) algorithm, evaluates the local density deviation of each data point in relation to its neighboring points to identify the anomalies. The sci-kit learn implementation was employed for this purpose. Among the nine available parameters, we focused on adjusting two key parameters: *n_neighbors* and *contamination*.

  For our specific implementation, we set the *n_neighbors* parameter to the default value of 20, while the *contamination* parameter was set to 0.1. The *contamination* rate represents the proportion of outliers in the dataset. Throughout our experimentation, we made significant observations and obtained noteworthy findings.

  A reduced *nneighbors* value, set at 10, improved the True Positive Rate (TPR) and anomaly detection by effectively identifying outliers with LOF scores different from 1, showcasing the algorithm's ability to discern anomalies from the majority of the data.

4.4 Explainability

Interpretability and explainability are essential in understanding classification models, as conventional models lack transparency in their decisions. The generation of rules, like if-then cases or hierarchical trees, aids in illuminating decision processes, enabling users to grasp the model's predictions and the underlying rationale.

Explainable Anomaly Detection (XAD) techniques can be classified into three categories: Pre-model, In-model, and Post-model [8]. Pre-model techniques involve feature selection and feature representation, and they operate solely on data without the use of any machine learning model. In-model techniques, on the other hand, utilize supervised or unsupervised models with built-in explainability, such as decision trees, which employ hierarchical trees to identify if-then cases and provide explanations for specific decisions. By focusing on the Post-model techniques, we aimed to gain a deeper understanding of the decision logic of our anomaly detection algorithms. Anchors generates simple and concise if-then rules that sufficiently explain the decisions made by the anomaly detection model. These rules are specific to local instances, meaning that even if a feature value changes, the predictions or rules remain mostly unchanged. This localized approach ensures robustness and stability in the explanations.

5 Results

Figure 5 presents a bar chart illustrating the comparison of recall scores among the three algorithms. Remarkably, the Local Outlier Factor exhibits subpar performance with a value of 0.08 on the Heart Disease dataset but surprisingly outperforms the other two algorithms on the Credit Card Fraud dataset with a value of 90%. This intriguing observation leads us to attribute the contrasting outcomes to the datasets varying densities, primarily stemming from class imbalance issues.

Conversely, the Isolation Forest secures the second position concerning recall scores on both datasets, and the obtained scores exhibit notable similarities which equals 45% and 44% on Heart Disease dataset and Credit Card Fraud dataset respectively. Interestingly, the One Class SVM's performance proves to be unsatisfactory on the Credit Card Fraud dataset which equals to 30%. Whereas the algorithm showcases a recall value of 44% on the Heart Disease dataset. We posit that this discrepancy occurs due to the One Class SVM's inability to effectively discern the class distributions of the anomalous instances and normal data points within this particular dataset.

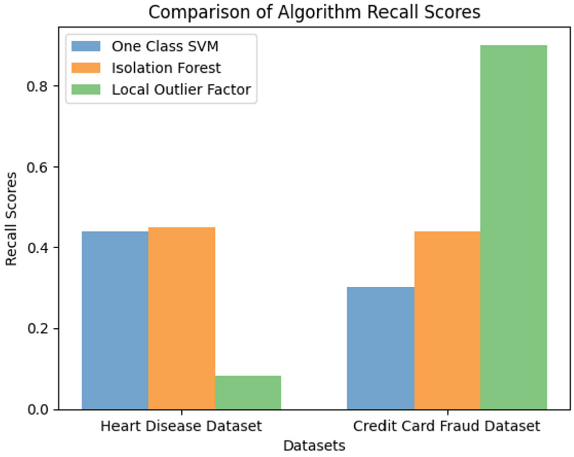


Fig. 5. Recall Score Comparisons

In Fig. 6, the bar chart illustrates the training times of the three algorithms. Notably, Local Outlier Factor demonstrates the shortest or nominal training duration on both datasets with values of 2.07 and 1.48 s for Heart Disease and Credit Card Fraud datasets respectively. This efficiency can be attributed to the fact that most of the computations in Local Outlier Factor occur during the prediction phase.

Isolation Forest ranks second in terms of training times on both datasets. However, it is worth mentioning that the training time for Isolation Forest is not constant and may vary across different runs on the same dataset with one being 46.57 and 35.89 s respectively for Heart Disease and Credit Card Fraud datasets. This variability could be attributed to the randomness involved in the feature selection process.

On the other hand, One Class SVM exhibits the longest training time on both datasets with values of 48.24 s in Heart Disease dataset and 57.42 s in Credit Card Fraud dataset. This is primarily due to the extensive number of calculations involved in defining the distance matrix and support vectors during the training phase. As a result, the computational complexity of One Class SVM contributes to its higher training time compared to the other two algorithms.

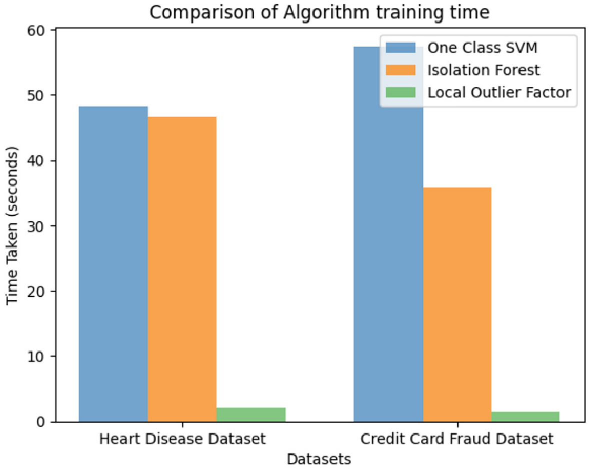
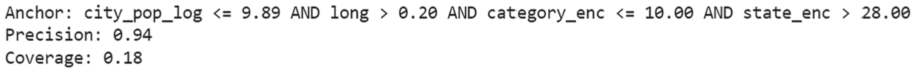


Fig. 6. The Time Taken for Training

5.1 Explainability Results

The generated anchors provide clear if-then rules based on specific features from the data that were used by the anomaly detection model to arrive at its decision. For instance, an anchor might state "if gender = Female and age < 20, then normal patient," or "if gender = male and age > 40, then check fitness." These rules are informative and human-readable, facilitating better understanding of the model's behavior.

Additionally, Anchors quantifies the accuracy and coverage of the generated rules, allowing users to gauge the reliability and scope of each explanation. This helps in assessing the trustworthiness of the rules and gaining insights into their impact on the model's predictions.



```
Anchor: city_pop_log <= 9.89 AND long > 0.20 AND category_enc <= 10.00 AND state_enc > 28.00
Precision: 0.94
Coverage: 0.18
```

Fig. 7. Rule generated by Anchors for Isolation Forest

Figure 7 shows the rules generated by Anchors for a trained isolation forest model. We input one instance to the anchors and it gives one rule, which provides explanation about why this particular instance is classified as anomaly or normal. Anchors take into consideration overall dataset and generates this rule. It also specifies the precision rate and the amount of instances covered. From the above generated rule, we see that the rule it is generated is 94 percent accurate.

Upon applying anchors to one class SVM, we observed that it effectively identifies the key features and their corresponding values associated with anomalous instances. The rules generated by this method shed light on the specific attributes that play a crucial role in the detection process. For example, Eq. 1 is the rule produced by Anchors for One Class SVM on the Heart Disease Dataset, achieving a precision of 0.82.

$$\begin{aligned}\text{Anchor: } & \text{ap\_lo} > 80.00\text{AND age\_years } \leq 46.22 \\ & \text{AND gender} > 1.00\text{AND weight } \leq 79.00 \\ & \text{AND height\_ft} > 5.58 \text{ AND active } \leq 1.00\end{aligned} \tag{1}$$

Conversely, when applying anchors to Isolation Forest, the generated rules appeared to be relatively shallow, often involving only one or two features. This outcome can be attributed to the intrinsic design of the Isolation Forest algorithm. Equation 2 is the rule generated by Anchors for Isolation forest on Heart Disease Dataset with a precision of 0.97

$$\text{Anchor : gluc} > 1.00 \text{ AND cholesterol} > 1.00 \tag{2}$$

6 Related Work

In this section, we present a review of the existing literature related to anomaly detection, aimed at identifying patterns or data points that deviate from the normal pattern within a dataset. Different approaches to anomaly detection, including supervised and unsupervised, have been explored in prior research, and various algorithms have been applied for outlier detection. While considering the refined problem statement of applying data mining techniques for anomaly detection, our goal is to implement and compare different effective methods for identifying anomalies that significantly deviate from the majority of the data.

The overview research done on the machine learning techniques for detecting the anomalies [11] helped us in narrowing to the domain of unsupervised techniques over the supervised ones with proper justifications. The supervised techniques require a significant amount of labelled data, where anomalies are explicitly identified and labelled. During the classification phase, the trained model is used to predict whether new instances are normal or anomalous by comparing them to the learned patterns. In [13], it is explained that the significant limitations of using the supervised techniques in anomaly detection. The important aspect being stated was that the acquiring of labelled data for anomalies can be challenging and time-consuming, especially in real-world scenarios where anomalies may be rare or evolving. Apart from that these techniques rely heavily on labelled training data, which makes them less effective in identifying anomalies that were not present in the training set.

Pointing down to the unsupervised algorithms, we understood that they make use of the unlabelled data where the anomalies are not predefined and align greater with the real-world scenarios of complex feature structures. These techniques focus on detecting instances that deviate significantly from the expected or normal patterns in the data. Therefore Unsupervised algorithms explore the data characteristics and identify anomalies based on their deviation from what is considered normal. The existing studies motivated us to focus reliably on the three algorithms which were specifically used for anomaly detection. They are:

1. Isolation Forest
2. One class support vector machine
3. Local Outlier Factor

In [9], the application of Isolation Forest detection to anomalies is discussed. Being an unsupervised model it builds an ensemble of isolation trees (iTrees) for a given dataset. The property of identifying the instances with short average path lengths on the isolation trees as anomalies becomes more peculiar to the algorithm. The algorithm which works on the principal idea of isolating anomalies rather than profiling normal instances has the ability to handle high dimensional datasets along with being computationally efficient and having interpretable results.

One class Support vector machine [14] gives significant insights into the pros of the algorithm in this specific context of anomalies. It captures the underlying structure of a target class and differentiates it from the rest of the data and also tries to find a function that is positive for regions with high density of points, and negative for small densities. The main advantage of the algorithm is the capability to handle non-linear relationships and imbalanced datasets.

Apart from these two algorithms, the work done by [1] helped us to delve more into the local outlier factor methods which have been specifically designed and contextually relevant in outlier or anomaly detection. It assigns each object in the dataset a degree of being outlier (Local Outlier Factor) and the data points with LOF values above a certain threshold are identified as Outliers We found the most significant advantage of the algorithm to be its usefulness for identifying anomalies in datasets with varying densities and complex geometric structures.

7 Conclusion

Our comprehensive study has yielded valuable insights into the suitability of diverse algorithms for varying dataset characteristics. Isolation Forests demonstrate robustness in handling static datasets with efficient training times and satisfactory performance. One Class Support Vector Machines (SVM) stand as a potent choice for well-separable datasets with substantial computational resources. Local Outlier Factor (LOF) excels in addressing datasets with fluctuating densities, showcasing its strength in scenarios marked by density variations. Each algorithm's distinct attributes render them well-suited to specific dataset traits, underlining the significance of tailored algorithm selection.

For future endeavors, delving deeper into the decision-making mechanisms of Isolation Forest and One Class SVM warrants exploration through the implementation of alternative rule generation systems like Scalable Bayesian Rule Lists, SHAP, hypercubes, among others. Such methods can provide a holistic comparison of these algorithms. Additionally, customizing rule generation with domain-specific insights holds potential for enhancing interpretability and relevance, further enriching our understanding of their performance.

References

1. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers (2000)
2. Chen, G., Zhang, X., Wang, Z.J., Li, F.: Robust support vector data description for outlier detection with noise or uncertain data. Knowl.-Based Syst. **90**, 129–137 (2015)
3. Degirmenci, A., Karal, O.: Efficient density and cluster based incremental outlier detection in data streams. Inf. Sci. **607**, 901–920 (2022)
4. Dey, A., Totel, E., Costé, B.: Daemon: dynamic auto-encoders for contextualised anomaly detection applied to security monitoring. In: Meng, W., Fischer-Hübner, S., Jensen, C.D. (eds.) ICT Systems Security and Privacy Protection - SEC 2022. IFIP Advances in Information and Communication Technology, vol. 648, pp. 53–69. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06975-8_4
5. Kartik: Fraud detection dataset (2020). What about this: https://www.kaggle.com/datasets/kartik2112/fraud-detection?select=fraudTrain.csv
6. Kavitha, M., Srinivas, P., Kalyampudi, P.L., Srinivasulu, S., et al.: Machine learning techniques for anomaly detection in smart healthcare. In: 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1350–1356. IEEE (2021)
7. Li, K.L., Huang, H.-K., Tian, S.F., Xu, W.: Improving one-class SVM for anomaly detection (2003)
8. Li, Z., Zhu, Y., van Leeuwen, M.: A survey on explainable anomaly detection, October 2022. http://arxiv.org/abs/2210.06959
9. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest, pp. 413–422 (2008). https://doi.org/10.1109/ICDM.2008.17
10. Manevitz, L.M., Yousef, M., Cristianini, N., Shawe-Taylor, J., Williamson, B.: One-class SVMs for document classification (2001)
11. Omar, S., Ngadi, M., Jebur, H., Benqdara, S.: Machine learning techniques for anomaly detection: an overview. Int. J. Comput. Appl. **79** (2013). https://doi.org/10.5120/13715-1478
12. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2019)
13. Rewicki, F., Denzler, J., Niebling, J.: Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series. Appl. Sci. (Switzerland) **13** (2023). https://doi.org/10.3390/app13031778
14. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Comput. **13**(7), 1443–1471 (2001). https://doi.org/10.1162/089976601750264965
15. Telo, J.: Ai for enhanced healthcare security: an investigation of anomaly detection, predictive analytics, access control, threat intelligence, and incident response. J. Adv. Anal. Healthc. Manag. **1**(1), 21–37 (2017)
16. ulianova, S.: Cardiovascular disease dataset (2019). https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset