Outlier Reporting By Inference on Turfs (ORBIT)

Henry Kasim

e0698342

University Supervisor:
Professor Wong Lim Soon

Industry Supervisor:
Dr. Lee Kee Khoon

DEPARTMENT OF COMPUTER SCIENCE NATIONAL
UNIVERSITY OF SINGAPORE July 25, 2023

Summary

This research introduces a novel outlier detection technique called Outlier Reporting By Inference on Turfs (ORBIT). It leverages the assumption that datasets predominantly consist of normal instances with only a few outliers. By selecting a small random sample of references that accurately represents the dataset's normal characteristics, ORBIT uses the distances between these reference points and their nearest neighbours to define the "turf" reachability boundary of each reference points; it is the regions of the datasets which are in the same clusters as these reference points. Those reference points with unusually large reachability boundary compared with other reference points are likely outlier themselves and are eliminated. The remaining (presumed normal) reference points are divided into random groups. A data instance is considered a local outlier with respect to a group if it falls outsides the turf reachability boundaries of all reference points in that group. By analyzing the distribution of the number of groups that the data instances occur as local outliers, ORBIT distinguishes true outliers from normal instances using the median plus *n* standard deviations of said distribution as a threshold.

ORBIT demonstrates outstanding performance, consistently outperforming other methods across various scenarios. Among the 19 datasets evaluated, ORBIT achieves the highest GMean score in 6 cases, while also obtaining close second or third best scores in 7 additional datasets. The GMean score, calculated as the geometric mean of sensitivity and specificity, serves as a comprehensive measure of the outlier detection method's effectiveness. With an average GMean score of 0.705, ORBIT surpasses its closest competitor by a margin of approximately 3 percentage points. Overall, this research provides valuable insights for implementing reliable and effective anomaly detection solutions.

Contents

Page

1 Introduction

1.1 Introduction and Background

Anomaly detection refers to the identification of deviations from the expected pattern or behavior, representing unusual or unexpected occurrences. In the realm of statistics, anomalies are observations that stand out from the rest of the dataset and are commonly referred to as outliers [1]. ACM (2023) places anomaly detection within the broader domains of computing methodology, machine learning, learning paradigms, and unsupervised learning. The primary objective of anomaly detection techniques is to identify potential cases or events that exhibit unusual behavior or deviate from the norm. These techniques typically operate without prior labeling of anomalous events and often rely on unsupervised learning approaches. Anomaly detection can be applied to diverse data types, including images, textual data, time-series data, and tabular data (Foorthuis, 2021). However, this report specifically concentrates on the analysis of tabular data. Tabular data is derived from observations of different subjects or entities at a specific point in time. Illustrative examples of tabular data encompass weather information for different cities, patients' medical records, and individual financial information within a population. Anomalies within tabular data manifest when an observation possesses distinctive characteristics or attributes compared to the remaining observations in the dataset.

There is a wide range of anomaly detection techniques available (Sikder and Batarseh, 2023; Samara et al., 2022; Smiti, 2020; Chandola, Banerjee, and Kumar, 2009). Within the area of unsupervised anomaly detection, several common categories can be identified:

[1] Outlier detection and anomaly detection are both techniques used in data analysis and machine learning to identify unusual patterns or data points in a dataset. However, there is a subtle difference between the two. Outliers are data points that deviate significantly from the rest of the data in a dataset. They are extreme values that lie far away from the majority of other data points. Outlier detection aims to identify these exceptional data points that do not conform to the general behavior of the data. Outliers can occur due to various reasons such as errors in data collection, measurement noise, or genuine rare events. Anomalies, on the other hand, refer to patterns in data that do not conform to the expected behavior or usual patterns in a given context. Unlike outliers, anomalies may not be extreme values; they can also be subtle deviations that are not necessarily statistically different from the rest of the data but are still considered unusual in the specific context of the problem. Anomaly detection is a broader concept that encompasses the identification of outliers as well as other unusual patterns in the data. It looks for data points or patterns that are rare, suspicious, or potentially indicative of some unexpected behavior. In summary, outlier detection specifically focuses on identifying extreme values that deviate significantly from the majority of data points, while anomaly detection is a broader concept that encompasses the identification of any unusual or unexpected pattern in the data, not just limited to extreme values. In this report, the aim is to detect not just data points that are extreme values, but also those that may represent subtle deviations in certain local regions of the data. The terminologies "outliers" and "anomalies" are used interchangeably in this context.

- Statistical methods: These techniques utilize statistical measures to identify deviations from expected behavior. They rely on statistical models to capture the underlying patterns of normal instances and flag observations that significantly deviate from these patterns. Examples of statistical methods include Median Standard Deviation, Mean Standard Deviation (Dave and Varma, 2014), Median Absolute Deviation (Leys et al., 2013), and Interquartile Range (Walfish, 2006).

- Clustering methods: This category of techniques groups similar observations together based on their inherent characteristics. Outliers are then identified as observations that do not conform to any specific cluster. By separating normal instances into distinct groups, clustering methods effectively detect anomalies as data points that do not belong to any defined cluster. Examples of clustering methods for anomaly detection include k-means (Chawla and Gionis, 2013) and MSD K-means (Wei et al., 2019).

- Nearest neighbor methods: These methods assess the distances between observations to identify anomalies. By comparing the dissimilarity of an instance to its nearest neighbors, these techniques can pinpoint anomalies that stand out due to their dissimilarity or abnormality. Examples of nearest neighbor methods include K-Nearest Neighbor (KNN) (Ramaswamy, Rastogi, and Shim, 2000), Local Distance-based Outlier Factor (LDOF) (Zhang, Hutter, and Jin, 2009), and Local Outlier Factor (LOF) (Breunig et al., 2000).

- Isolation methods: Isolation techniques construct models that define the normal behavior of the data. They isolate anomalies as instances that deviate significantly from these models. By building boundaries around normal instances, isolation methods effectively identify anomalies that fall outside these boundaries. Examples of isolation forest methods include Isolation Forest (IF) (Liu, Ting, and Zhou, 2008), extended Isolation Forest algorithm (EIF) (Hariri, Kind, and Brunner, 2019), and cluster-based Isolation Forest (Shao et al., 2022).

To evaluate the performance of anomaly detection techniques, various evaluation metrics can be utilized. These metrics provide quantitative measures of a technique's effectiveness in detecting anomalies. Commonly employed metrics include accuracy, precision, F1-score, specificity, sensitivity, and geometric mean (Umer et al., 2022; Dahmen and Cook, 2021; Estiri, Klann, and Murphy, 2019; Ngo and Veeravalli, 2015). Additionally, metrics based on ROC curves (Receiver Operating Characteristic) and confusion matrices are frequently employed, offering insights into the true positive rate, false positive rate, and overall accuracy of anomaly detection (Umer et al., 2022).

1.2 Research Motivation and Considerations

Understanding various anomaly detection techniques is essential to determining their effectiveness in detecting anomalies accurately. This research aims to explore various anomaly detection techniques to identify the aspects that are efficient and effective. By examining different facets of anomaly detection, valuable insights can be gained to implement reliable solutions. The motivations for achieving reliable anomaly detection performance can be categorized into domain-related, data-related, model-related, and result-related factors. Understanding and addressing these motivations contribute to the overall success of anomaly detection performance.

1.2.1 Domain-Related Motivation

Domain-related motivations for implementing anomaly detection techniques vary across industries, addressing unique requirements and achieving key objectives. Anomaly detection serves as a crucial tool in each domain, addressing specific needs and overcoming challenges.

In the **manufacturing**, the motivation behind anomaly detection lies in the pursuit of operational excellence and efficiency. Manufacturers are dedicated to identifying anomalies that can impact production processes, product quality, and equipment reliability. Real-time detection of deviations from expected patterns enables proactive maintenance, minimizes downtime, and ensures consistent product quality. This timely anomaly detection empowers manufacturers to optimize production workflows, reduce waste, and enhance overall operational performance. Anomalies in manufacturing can arise from equipment failures, device malfunctions, and quality control discrepancies. To identify these anomalies, specialized models are required to uncover subtle deviations in behavioral patterns. By detecting anomalies, manufacturers can proactively address issues, prevent production failures, and improve overall quality.

Anomalies in **healthcare** can manifest in deviations from normal patterns that are associated with good health conditions. Analyzing vast volumes of historical observations using anomaly detection techniques empowers medical professionals to make informed decisions. By identifying anomalies in medical data, healthcare providers can detect diseases at an early stage, improve patient outcomes, and enhance overall healthcare management.

In the **cybersecurity** domain, the motivation for anomaly detection arises from the increasing sophistication of cyber threats, coupled with the imperative to safeguard sensitive data and systems. The primary objective of cybersecurity is to promptly identify suspicious network activities, unauthorized access attempts, and anomalous user behavior. Real-time detection and rapid response mechanisms are essential to detect anomalies that indicate potential security breaches. Anomaly detection techniques help in detecting unusual patterns of network activities, enabling organizations to take immediate action to mitigate cyber threats and protect sensitive information.

Anomalies in the **financial** sector often take the form of fraudulent activities. Detecting anomalies and early indicators of potential illicit transactions is crucial for proactive fraud detection and protection. Anomaly detection techniques can identify suspicious financial transactions, unusual patterns in customer behavior, or potential fraud schemes, enabling financial institutions to prevent financial losses and maintain trust with their customers.

In summary, domain-related motivations for anomaly detection are driven by the specific requirements and challenges within each industry. Successful implementation of anomaly detection techniques requires a deep understanding of the unique characteristics and requirements within each industry. By customizing approaches to cater to specific industries, anomaly detection can attain optimal performance, offer invaluable insights, and facilitate well-informed decision-making.

1.2.2 Data-Related Considerations

Data-related motivations in outlier detection revolve around the critical role of data quality and integrity in achieving effective anomaly detection. Ensuring clean, accurate, and reliable datasets is essential for accurate identification of anomalies (Chandola, Banerjee, and Kumar, 2009). The data-related considerations in outlier detection include the following:

Comprehensive Understanding of Data: Before performing anomaly detection, it is essential to have a comprehensive understanding of the data. This includes evaluating important aspects such as data size, type, distribution, and patterns. By understanding these characteristics, researchers can choose appropriate anomaly detection techniques and set suitable parameters for detection.

Identifying Data Quality Issues: The evaluation process helps in identifying data quality issues such as missing values, duplicates, and inconsistencies. These issues can adversely affect the accuracy of anomaly identification. By detecting and addressing data quality issues through preprocessing and cleaning steps, the reliability and effectiveness of anomaly detection can be improved.

Handling Data Heterogeneity: Data heterogeneity refers to differences in feature scales or distributions within the dataset. Anomaly detection techniques need to consider and handle such heterogeneity appropriately. Data preprocessing steps, like normalization or standardization, can be applied to address this issue and ensure that the detection algorithm performs optimally (Kandanaarachchi et al., 2020).

Balancing Data Preprocessing: While data preprocessing is essential, it is worth noting that not all anomaly detection techniques require extensive preprocessing. In some cases, inappropriate preprocessing measures can even undermine the performance of anomaly detection. Therefore, a deep understanding of the underlying anomaly detection technique and its relationship to data preprocessing is necessary to strike the right balance.

Consideration of Data Processing Steps: When selecting suitable anomaly detection techniques, careful consideration should be given to various data processing steps. For example, imputation techniques can be applied to handle missing values, and normalization or scaling methods can be used to handle data heterogeneity. The choice of appropriate data processing steps depends on the specific characteristics of the dataset and the requirements of the anomaly detection technique being employed.

In summary, a comprehensive understanding of the data, including its size, type, distribution, and quality, is crucial for effective anomaly detection. The evaluation process helps identify data-related issues, and appropriate data preprocessing steps can be applied to address these issues. However, it is important to strike a balance and choose preprocessing measures that align with the anomaly detection technique being used to avoid any negative impact on detection performance.

1.2.3 Model-Related Considerations

In anomaly detection, the choice of appropriate models is crucial for achieving accurate and effective anomaly detection. The motivation behind model selection revolves around improving detection performance, scalability, interpretability, and adaptability to different types of data. Considerations related to model selection include the following:

Choosing the Appropriate Technique: The effectiveness of anomaly detection relies on selecting the most suitable technique for the specific task and available resources. Different techniques have varying strengths and weaknesses, and their performance can vary depending on the characteristics of the data and the anomalies being targeted. Therefore, understanding the capabilities and limitations of each technique is crucial for making an informed choice.

Understanding and Adjusting Model Parameters: Within each technique, there are parameters that require careful comprehension and adjustment to optimize the performance of anomaly detection. These parameters control the behavior of the model and can significantly impact its ability to detect anomalies accurately. It is important to understand the implications of each parameter and adjust them based on the specific requirements of the task and the characteristics of the data.

Parameter Sensitivity to Anomaly Types: Certain parameters in anomaly detection techniques may perform well in detecting anomalies with specific characteristics but may be less effective in identifying anomalies of different types. For example, in the case of the nearest neighbor approach, the choice of the parameter on the number of nearest neighbors considered can impact the detection of densely clustered anomalies. A lower value for this parameter may encounter difficulties in detecting anomalies that are densely packed or closely located.

Evaluation of Model Performance: To assess the effectiveness of an anomaly detection technique, it is essential to evaluate its performance using appropriate evaluation metrics. This evaluation helps in understanding how well the model performs in detecting anomalies and provides insights into areas where improvements or adjustments may be needed.

Model Selection and Combination: In some cases, a single anomaly detection technique may not be sufficient to capture all types of anomalies or handle complex scenarios. Model selection and combination techniques, such as ensemble methods or hybrid approaches, can be employed to improve overall performance and increase the robustness of the detection system.

In summary, model-related considerations in outlier detection involve selecting the appropriate technique, understanding and adjusting model parameters, considering the sensitivity of parameters to different anomaly types, evaluating model performance, and potentially employing model selection and combination strategies. By addressing these issues, the effectiveness and efficiency of anomaly detection can be enhanced, leading to more accurate identification of anomalies in various applications.

1.2.4 Result-Related Considerations

To accurately evaluate the effectiveness and performance of an anomaly detection technique, careful selection of appropriate performance evaluation metrics is crucial. These metrics should align with the objectives and requirements of the application domain, providing valuable insights into the method's performance.

The choice of suitable performance evaluation metrics is driven by several key factors. Firstly, anomalies are often rare occurrences, leading to imbalanced datasets with a large number of normal instances and only a few anomalies. Traditional metrics like accuracy or error rate can be misleading in such imbalanced scenarios and fail to effectively reflect the method's performance (Umer et al., 2022).

Secondly, the costs and impacts of false positives and false negatives can vary significantly depending on the application domain. For example, in cybersecurity, a false negative (failure to detect an actual intrusion) can have severe consequences, while a false positive (identifying a benign activity as anomalous) may be less critical (Chalapathy and Chawla, 2019). In healthcare, labeling a healthy individual as anomalous (false positive) can result in unnecessary testing or treatment, while failing to detect a disease (false negative) can have serious repercussions (Chandola, Banerjee, and Kumar, 2009).

Additionally, other important factors need to be considered when evaluating the performance of the method. This includes the speed of anomaly detection, which is influenced by the specific requirements of the application domain. For example, immediate anomaly detection is crucial in cybersecurity, while in a manufacturing context, it may be of lesser importance (Angin, Bhargava, and Ranchal, 2019). Furthermore, human factor requirements, such as providing explanations or confidence levels for detected anomalies, play a significant role in enhancing human understanding of the anomaly detection method (Siddiqui et al., 2019).

The selection of appropriate performance evaluation metrics should be a thoughtful and comprehensive process, considering the specific application domain, data characteristics, and the objectives of the anomaly detection method. By choosing the right evaluation metrics, the goal is to accurately assess the performance of the method and gain meaningful insights into its effectiveness.

1.3 Research Contribution

The main research contribution of this thesis is the development of a novel outlier detection technique named Outlier Reporting By Inference on Turfs (ORBIT). ORBIT utilizes three reasonable assumptions to develop a simple outlier detection approach:

- Assumption 1: There is a large number of normal instances and only a few outliers. Consequently, when selecting a small random sample (referred to as a set of references), there is a high probability that it will contain few or no outliers.
- Assumption 2: Normal samples tend to be closer to other normal samples than to outliers. Based on this, along with the first assumption, the nearest neighbors of the randomly selected references are more likely to be normal rather than outliers. Thus, these neighbors and their distances from the reference can be used to determine the typical distances of normal samples from this reference. As a practical rule-of-thumb, a high-confidence boundary for a normal region or a portion of it can be defined as the radius of the median plus n standard deviations of these neighbor distances. The region within this boundary is referred to as the "turf" of the reference.
- Assumption 3: The references can be randomly split into several tens of groups. If a group is comprised of references which are well spread-out, the combined turfs of its references will have a good chance of covering most normal regions. By using a sufficient number of reference groups, all normal regions will be covered, albeit not by any single group alone. Consequently, a normal instance may fall outside the turf of multiple groups but should not be outside too many groups (as a normal instance must be inside at least one normal region). This, combined with the second assumption that outliers are further away from normal instances, leads to a simple rule for distinguishing outliers from normal instances: "persistent outliers" (instances outside the turfs of most, if not all, groups) are considered outliers. Therefore, the distribution of the number of times a data instance falls outside the turf of a group can be analyzed. Since, by the first assumption, most data instances are normal, a majority of this distribution corresponds to normal instances, while the extreme right end corresponds to outliers. As a practical rule of thumb, the threshold for defining the extreme right end of the distribution can be set as the median plus n standard deviations.

The key results from the extensive experiments and evaluations conducted on ORBIT demonstrate its performance and robustness. The evaluations covered various factors, such as parameter settings, handling of irrelevant features, data normalization, high-dimensional data, and benchmark dataset performance. Across the 19 evaluated benchmark datasets, ORBIT consistently outperformed other anomaly detection methods, achieving the highest GMean score on 6 datasets and an average GMean score of 0.705. The closest competitor, the ROD method, achieved an average GMean score of 0.688 and obtained the highest score on 2 datasets.

Upon further investigation on an ORBIT's lower performance dataset, shuttle dataset, it reveals that the data contained a significant number of densely clustered outlier instances exceeding the specified parameter for the number of nearest neighbors. However, by adjusting a parameter according to the high proportion of outliers, ORBIT's performance greatly improved. ORBIT's GMean score surged to 0.970, comparable to the score of 0.976 which was the highest GMean score of other approaches. This successful adaptation highlights ORBIT's flexibility and its capability to outperform alternative methods in outlier detection tasks.

These results highlight ORBIT's effectiveness in practical applications, demonstrating its superiority in accurately identifying outliers and surpassing competing approaches. ORBIT's performance, adaptability, and robustness make it a valuable tool for anomaly detection in various domains and datasets.

2 Literature Review

2.1 Data Characteristics

To effectively apply anomaly detection techniques, it is essential to have a comprehensive understanding of the data characteristics. This section explores the existing literature on data characteristics and their significance in anomaly detection. Several key challenges related to data characteristics include data distribution and patterns, data quality, data heterogeneity, and the size of the data. These factors directly influence the selection of appropriate anomaly detection techniques (Kandanaarachchi et al., 2020).

To ensure a thorough understanding of the data before performing anomaly detection, several steps can be taken, starting with data exploration. Data exploration involves carefully examining the dataset to gain insights into its structure, size, and features. Visualizations such as histograms, scatter plots, and box plots can be employed to visualize the data and identify patterns or irregularities (Seo and Shneiderman, 2004). Through this exploratory journey, researchers and practitioners can detect missing values, outliers, and other data quality issues that may hinder the anomaly detection process.

By understanding the data characteristics through techniques like data exploration, researchers and practitioners acquire the necessary insights to make informed decisions when selecting suitable anomaly detection methods. These chosen methods are aligned with the specific data characteristics and effectively address the challenges inherent in the data (Han et al., 2022). This comprehensive understanding establishes the foundation for accurate and effective anomaly detection.

In summary, having a comprehensive understanding of data characteristics is useful for effective anomaly detection. Exploring the data through techniques like data exploration enables researchers and practitioners to identify patterns, anomalies, and data quality issues. This understanding guides the selection of appropriate anomaly detection methods that align with the specific data characteristics, leading to accurate and effective anomaly detection.

2.1.1 Data Quality and Heterogeneity

In the domain of anomaly detection, research has been conducted to investigate the significance of data quality. Poon et al. (2021) have identified several challenges related to data quality, including data entry errors, measurement errors, distillation errors, and data integration errors. These challenges can arise from human involvement, physical measurement processes, pre-processing and summarization of raw data, and the integration of data from multiple sources.

Ensuring data quality in anomaly detection has been emphasized by Wu and Keogh (2021) and Kandanaarachchi et al. (2020). In the context of outlier detection, it is essential to have a dataset with a small proportion of outliers when evaluating anomaly detection techniques. Data normalization plays a vital role in anomaly detection as it brings features to a consistent scale, enabling the detection of anomalies based on their deviation from the norm rather than their absolute values. Kandanaarachchi et al. (2020) highlight that the choice of normalization method can significantly impact the performance of outlier detection techniques, creating an interplay between the dataset and the sensitivity of the detection techniques to normalization.

Examining the literature on normalization approaches in intrusion detection and the impact of normalization on benchmark datasets (Umar and Chen, 2020; Kandanaarachchi et al., 2020), two common data normalization techniques are Minimum and Maximum normalization and Mean with Standard Deviation normalization. Minimum and Maximum normalization scales the data to a fixed range between 0 and 1, preserving the original distribution. However, it is sensitive to outliers and can be skewed by extreme values. On the other hand, Mean with Standard Deviation normalization, or z-score normalization, rescales the data to have a mean of 0 and a standard deviation of 1. It maintains the shape of the distribution and allows for the comparison of variables measured on different scales. Nevertheless, z-score normalization is also sensitive to outliers and may distort the distribution, making it less appropriate for non-normally distributed data.

Considering the impact of data quality and the choice of normalization techniques is crucial in anomaly detection, as it directly influences the performance and reliability of detection methods. By addressing data quality issues and employing appropriate normalization techniques, researchers can enhance the performance and interpretability of anomaly detection results.

2.1.2 Data Size

Research and proposed techniques in anomaly detection have addressed the challenges posed by big data, specifically in terms of high instance numbers and high dimensionality. In a comprehensive survey by Thudumu et al. (2020), various anomaly detection methods were evaluated for their performance in high-dimensional big data scenarios. The survey highlighted the limitations of traditional approaches, discussed current strategies such as parallelization and distributed computation, and reviewed challenges associated with handling high-dimensional big data.

One approach, as presented by Kamalov and Leung (2020), tackles high dimensionality through Principal Component Analysis (PCA) for dimensionality reduction and Kernel Density Estimation (KDE) for outlier detection. PCA reduces dimensionality, addressing sparsity issues, while KDE models density distribution to identify outliers based on their low probability. This two-step approach excels in outlier detection, outperforming benchmark methods in terms of F1-score and demonstrating superior computational efficiency.

Riahi-Madvar et al. (2021) propose an efficient method for outlier detection in high-dimensional data by introducing the Maximum-Relevance-to-Density (MRD) and minimum-Redundancy-Maximum-Relevance-to-Density (mRMRD) algorithms. MRD employs mutual information to select informative features, while mRMRD considers feature redundancy for more efficient subspace selection. The selected subspace is used to compute the outlierness of data points using the Local Outlier Factor measure. Experimental results highlight the improved accuracy, reduced computational complexity, and higher accuracy with increasing dimensionality achieved by the proposed algorithms.

In summary, developing effective methods to address high dimensionality big data is crucial to ensure the performance and reliability of anomaly detection in real-world applications. By addressing the challenges posed by high dimensionality big data, researchers and practitioners can enhance the robustness and effectiveness of anomaly detection systems in analyzing large-scale and complex datasets.

2.1.3 Nature of the Anomaly

Understanding the nature of anomalies is essential for effective anomaly detection, providing insights into their defining characteristics and various types present in datasets (Foorthuis, 2021). Foorthuis (2021) conducted a comprehensive literature review, resulting in a typology of data anomalies that encompasses dimensions such as data type, cardinality of relationship, anomaly level, data structure, and data distribution. This research categorizes anomaly types into univariate and multivariate anomalies for qualitative and tabular data. Univariate anomalies include extreme tail values and isolated intermediate values, focusing on instances significantly different from the rest of the dataset based on a single feature. Multivariate anomalies encompass peripheral points and enclosed points, where instances differ from the dataset in multiple feature values. Local density anomalies and global density anomalies represent atomic multivariate data, denoting instances considered anomalous in relation to dataset density. Additionally, point-based aggregate anomalies and distribution-based aggregate anomalies are identified as aggregate multivariate anomalies, with the former related to the aggregate spatial location of instances and the latter associated with the anomalous distribution of instances.

Goldstein and Uchida (2016) identified three types of anomalies that can be detected in datasets. Global anomalies refer to instances that significantly differ from the majority of data points in terms of their attributes. Local anomalies are instances considered anomalous only when compared to their nearby neighborhood. Micro-cluster anomalies pertain to groups of instances that exhibit differences from the normal instances.

Broadly categorized, anomalies can be classified based on attribute values (spatial location) and density. Attribute-based anomalies are classified as either local or global anomalies. Local anomalies are instances that deviate from their neighboring instances or a small local region, while global anomalies deviate from the entire dataset or larger regions. Global anomalies are typically easier to detect than local anomalies, as local anomalies can be concealed within the normal behavior of the surrounding data. Density-based anomalies are categorized as sparse or dense anomalies. Sparse anomalies refer to outliers that are spread out or distant compared to normal instances, while dense anomalies consist of outlier instances that are closer together but occur in smaller numbers relative to normal instances. The combination of attribute values and density yields categories such as global sparse outliers, global dense outliers, local sparse outliers, and local dense outliers.

Understanding the nature of anomalies is crucial in anomaly detection as it provides a fundamental understanding of their defining characteristics and the different types present in datasets. This understanding is key for developing highly effective anomaly detection algorithms that have the capability to detect a wide range of anomaly types, thereby enhancing the overall performance and reliability of anomaly detection systems.

2.2 Model: Unsupervised Anomaly Detection Approaches

In the field of anomaly detection, numerous studies have extensively explored different techniques, leading to the classification of approaches into statistical, classification, clustering, spectral, nearest neighbor distance, and isolation-based methods (Sikder and Batarseh, 2023; Samara et al., 2022; Smiti, 2020; Chandola, Banerjee, and Kumar, 2009). This report focuses on unsupervised anomaly detection techniques, excluding the exploration of classification and spectral approaches. The subsequent subsections dwell into a detailed description of selected anomaly detection methods, highlighting their strengths and limitations. The evaluation of these approaches considers data characteristics and various types of anomalies, with a particular emphasis on cross-sectional data.

2.2.1 Statistical Approach

The statistical approach to anomaly detection leverages the statistical properties of the data to identify instances that deviate from the expected distribution. This method assumes that normal data instances follow a known statistical distribution, such as Gaussian or Poisson, and detects anomalies by identifying instances that deviate from this distribution. Several statistical approaches are commonly used for anomaly detection, including Median Standard Deviation, Mean Standard Deviation (Dave and Varma, 2014), Median Absolute Deviation (Leys et al., 2013), and Interquartile Range (Walfish, 2006).

Median and Mean Standard Deviation measure the spread of data by calculating the middle point or average and the standard deviation (Dave and Varma, 2014). A threshold value is then derived based on a certain number of standard deviations away from the middle point or average. In outlier detection, Median Standard Deviation is preferred over Mean, as the median value is generally not influenced by extreme values in the data. This method can be used to establish lower and upper bound thresholds for identifying outliers. Data points that fall outside a certain number of standard deviations away from the median are considered outliers. The sensitivity of outlier detection can be controlled by adjusting the number of standard deviations away from the median. Another method, Median Absolute Deviation (MAD) (Leys et al., 2013), measures the variability of a dataset based on the median of the absolute deviations from the median. The use of absolute deviation makes MAD a robust measure less affected by outliers compared to standard deviation. Interquartile Range (IQR) is a statistical dispersion measure that describes the range between the first quartile (25th percentile) and the third quartile (75th percentile) of a dataset (Walfish, 2006). IQR can be used as a threshold for identifying outliers. Data points that fall below the first quartile minus 1.5 times the IQR or above the third quartile plus 1.5 times the IQR are typically considered outliers.

The main advantage of the statistical approach is its ease of understanding in terms of the approach and detection results. For example, in the Median Standard Deviation method, outliers are identified as instances that are a certain number of standard deviations away from the median. Furthermore, the evaluation of outlier detection is based on the univariate nature of the data, making it visually presentable and enhancing the understanding of the detection results. However, the statistical approach may not be as effective as more sophisticated methods, especially when dealing with datasets containing complex patterns, distributions, or multiple types of outliers. Additionally, the choice of the standard deviation threshold value can significantly impact the performance of the method. Another limitation of the statistical approach is that the described methods evaluate each feature independently (univariate analysis) and assume that outlier points are located outside the assumed unimodal distribution (global outliers). Hence, selecting the best statistical approach for outlier detection may not be straightforward due to these limitations (Boushey et al., 1995).

In conclusion, while the statistical approach offers simplicity and interpretability, it may not be the most suitable choice for detecting anomalies in datasets with intricate patterns or diverse types of outliers. Careful consideration should be given to the limitations of the statistical methods, such as the reliance on univariate analysis and assumptions about the distribution of outliers, when choosing the appropriate approach for anomaly detection.

2.2.2 Nearest Neighbour Approach

The nearest neighbor approach in anomaly detection involves analyzing the proximity of data observations to their neighboring instances. This can be done by measuring the distance between an observation and its K nearest neighbors or by evaluating the relative density of the observation's neighborhood. Some distance-based methods include K-Nearest Neighbor (KNN) (Ramaswamy, Rastogi, and Shim, 2000) and Local Distance-based Outlier Factor (LDOF) (Zhang, Hutter, and Jin, 2009). One density-based method is the Local Outlier Factor (LOF) (Breunig et al., 2000).

The KNN method, proposed by Ramaswamy, Rastogi, and Shim (2000), detects outliers by ranking points based on their distance to their K nearest neighbors. The top N points in this ranking are considered outliers. To address outlier detection in scattered data, Zhang, Hutter, and Jin (2009) introduced LDOF, which determines the degree to which a point differs from its neighborhood. It assigns a score to each point based on its dissimilarity from neighboring points, and the points with the highest scores are identified as outliers. LOF, proposed by Breunig et al. (2000), computes the local density of each data point and compares it with the density of its neighboring points to assign an LOF score, indicating the degree of abnormality compared to its neighbors.

One advantage of the nearest neighbor approach is that it does not rely on assumptions about the data distribution. It excels at detecting global and sparse anomalies as well as local anomalies based on density. However, there are limitations to consider. Anomalies with a sufficient number of close neighbors may be inaccurately predicted or missed. Additionally, the computational complexity of the approach can be high, particularly when computing the nearest neighbors for all observations.

In conclusion, the nearest neighbor approach offers flexibility by not assuming any specific data distribution and has the capability to detect various types of anomalies. Its strengths lie in detecting global, sparse, and local anomalies. However, it may struggle with anomalies having a large number of close neighbors and can be computationally intensive. Choosing appropriate parameter values, such as K and distance measure, is crucial for its effectiveness. Therefore, when utilizing the nearest neighbor approach, careful consideration should be given to its limitations and parameter selection to achieve accurate and efficient anomaly detection.

2.2.3 Clustering Approach

The clustering approach aims to group similar observations into the same clusters, and it can be categorized into several methods. One method identifies outliers as instances that are far away from their cluster centroid, improving computational performance by comparing instances only to the representative instance within each cluster (Chandola, Banerjee, and Kumar, 2009). Another method identifies outliers as instances belonging to smaller groups compared to larger normal groups (Ahmed, Mahmood, and Islam, 2016). A third method detects outliers as instances with a different distribution or area size compared to other groups (Chandola, Banerjee, and Kumar, 2009). The strengths of the second and third methods lie in their ability to improve outlier detection performance for dense or sparse anomalies by evaluating the number of instances and their distribution within each cluster.

Among the clustering-based methods for outlier detection, the k-means clustering algorithm is commonly employed (Chawla and Gionis, 2013). This algorithm selects k points as the initial cluster centers, ranks all points by their distance to their nearest cluster center, and assigns each point to its nearest cluster; then, the new cluster centers are determined, and the process iterates until a stable solution is reached. If the distance of a point to its cluster center exceeds a threshold, it is regarded as an outlier. Another proposed method, MSD K-means (Wei et al., 2019), combines the Mean Standard Deviation (MSD) statistical approach with the K-means algorithm to detect both global and local outliers. The

MSD algorithm eliminates extreme values that act as global outliers, while removing noisy data points enables the K-means algorithm to achieve optimal local clusters.

The clustering approach offers several strengths in detecting anomalies. It effectively identifies sparse anomalies that do not belong to any cluster and can detect anomalies that are far away from clusters despite being surrounded by normal observations (local outliers). Moreover, the approach is adaptable to various data distributions and types, as different clustering algorithms can be employed. It is also capable of detecting outliers that are sparse and surrounded by normal observations. However, the computational complexity of the clustering approach depends on the chosen clustering algorithm, and some methods may incorrectly assign anomaly observations to clusters. Additionally, the approach may struggle in identifying densely-packed anomalies, such as repeated network intrusions. Furthermore, for datasets with complex geometric shapes, defining meaningful clusters can be challenging, making the clustering approach less suitable for such scenarios.

In conclusion, the clustering approach for anomaly detection offers several strengths, including its ability to detect sparse anomalies, local outliers, and anomalies that do not conform to any existing clusters. It is adaptable to various data distributions and types and can effectively handle outliers that are surrounded by normal observations. However, the clustering approach has limitations, such as struggling to identify densely-packed anomalies and datasets with complex geometric shapes. There is also a risk of incorrect assignment of anomalies to clusters, and the computational complexity depends on the chosen clustering algorithm. Despite these limitations, the clustering approach provides valuable insights and can be a powerful tool in detecting anomalies in different types of data.

2.2.4 Isolation Approach

Introduced in 2008, the Isolation Forest (IF) (Liu, Ting, and Zhou, 2008) has since been extended and improved through various techniques (Chabchoub et al., 2022; Hariri, Kind, and Brunner, 2019). One such extension is the Extended Isolation Forest algorithm (EIF) (Hariri, Kind, and Brunner, 2019), which isolates observations based on hyperplanes to avoid vertical and horizontal feature splits. A more recent extension is the cluster-based Isolation Forest (Shao et al., 2022), which addresses the limitation of local outlier detection by first applying a clustering method to all observations and then calculating the anomaly score for each observation within the clusters.

The anomaly detection approach of the Isolation Forest is based on the concept that anomaly observations can be rapidly isolated within binary tree splits. The method excels in detecting global outliers and exhibits efficient computational performance. This performance is achieved by randomly selecting subsets of instances and features and combining the results from these subsets. However, a primary limitation of the Isolation Forest approach lies in its inability to detect anomalies located within the normal observations, also known as local anomalies. In cases where the data follows an exponential distribution, the Isolation Forest may generate numerous false positives, as it mistakenly identifies normal observations at the tail end of the distribution. Additionally, the accuracy of anomaly detection decreases as the number of irrelevant features increases (Liu, Ting, and Zhou, 2008).

In conclusion, the Isolation Forest is a powerful anomaly detection method that shows strengths in detecting global outliers and offers efficient computational time performance. However, it struggles with local anomaly detection, particularly in cases where the data follows an exponential distribution. The accuracy of the Isolation Forest also diminishes with an increasing number of irrelevant features. Awareness of these limitations is crucial when applying the Isolation Forest technique, and researchers should consider alternative approaches or modifications when dealing with local anomalies or datasets with numerous irrelevant features.

2.2.5 More Recent Approaches

This subsection explores several recent implementations of anomaly detection methods. One such implementation is the Histogram-Based Outlier Score (HBOS) proposed by Goldstein and Dengel (2012). HBOS creates univariate histograms for each feature dimension, representing density estimations of data points. By computing the HBOS score based on the heights of bins where each instance is located, it efficiently captures global anomalies across multiple dimensions, identifying outliers that are globally distinct in the dataset.

Another approach, Angle-based Outlier Detection (ABOD), was introduced by Kriegel, Schubert, and Zimek (2008). ABOD calculates angles formed by each data point (pivot) with all other data pairs in the dataset. The variance of these angles is then computed, and values below a specified threshold are identified as potential anomalies. A low variance indicates a likely anomaly that lies on one side of normal data points, whereas a high variance suggests a normal data point that is surrounded in all directions by many other normal data points. ABOD is effective in capturing anomalies that deviate significantly from the normal patterns in the dataset based on the angles formed between data points.

A novel implementation is Rotation-based Outlier Detection (ROD), proposed by Almardeny, Boujnah, and Cleary (2020). ROD decomposes the attribute space into 3D-subspaces, represented by 3D-vectors. These vectors are then rotated around the geometric median using the Rodrigues rotation formula to construct the overall outlying score. ROD does not rely on any specific data distribution assumptions, making it versatile and capable of capturing anomalies in complex and diverse datasets.

Another approach is Copula-Based Outlier Detection (COPOD) introduced by Li et al. (2020). COPOD draws inspiration from copulas, statistical methods used to model multivariate data distributions. It constructs an empirical copula to capture dependencies between variables and estimates the tail probabilities of each data point in the dataset. By utilizing copulas, COPOD effectively captures complex dependencies and interactions between variables, enabling it to detect outliers based on their extreme behavior compared to the rest of the data. This approach offers a flexible and powerful way to identify outliers in multivariate datasets, making it valuable for anomaly detection tasks across various domains.

Lastly, Empirical-Cumulative-Distribution-based Outlier Detection (ECOD), proposed by Li et al. (2022), focuses on outliers representing "rare events" found in the tails of distributions. ECOD estimates the underlying data distribution through empirical cumulative distributions computed per dimension. By estimating the tail probabilities for each data point along each dimension and aggregating them across all dimensions, ECOD determines the outlier score for each data point. While ECOD performs well in datasets with unimodal distributions, it may be less effective in multimodal datasets. Nevertheless, its reliance on empirical distributions and tail probabilities makes ECOD a valuable approach for identifying rare events and outliers in unimodal datasets.

2.3 Result: Evaluation Metrics

In order to accurately assess the effectiveness of anomaly detection techniques, it is crucial to evaluate them by comparing their performance in detecting known anomalies. However, selecting appropriate performance evaluation metrics in outlier detection is motivated by several factors (Umer et al., 2022; Estiri, Klann, and Murphy, 2019; Ngo and Veeravalli, 2015). One factor is the rarity of anomalies, which often leads to heavily imbalanced datasets. Another factor is that the consequences of false positives and false negatives should have similar impacts.

In the domain of intrusion detection for industrial control systems, Umer et al. (2022) have evaluated various research works and summarized the evaluation metrics used in this area. These metrics include accuracy, precision, recall, F-measure, receiver operating characteristic (ROC), and area under the ROC curve (AUC) as shown in Figure 2.1. Accuracy measures the ratio of correctly predicted instances to the total number of instances in the dataset. Accuracy in anomaly detection can be misleading as it is strongly influenced by the ratio of outliers to non-outliers in the evaluation set. If the dataset contains an imbalanced distribution, where the number of non-outliers outweighs the number of outliers, the accuracy measure may be skewed towards the majority class and not accurately represent the performance of the anomaly detection model. Precision measures the ratio of correctly identified positive instances to all instances predicted as positives, indicating a low false positive rate or fewer false alarms. Precision, much like accuracy, is also affected by the ratio of outliers to non-outliers in the evaluation set. For example, if the proportion of outliers versus non-outliers in the evaluation set differs greatly from the proportion in the real world, the precision observed from the evaluation set would be very different from the precision in the real world, rendering the former a useless or misleading expectation of how the outlier detection method would perform in the real world. Consequently, relying solely on precision may not offer a good assessment of the model's performance. Recall measures the ratio of true positives to the sum of true positives and false negatives, indicating the model's ability to detect a high number of anomalies. The F-measure combines precision and recall into a single metric using the harmonic mean, as both measures are important in intrusion detection.

The ROC curve is based on the true positive rate (TPR) plotted against the false positive rate (FPR).

AUC (Area Under the Curve) represents the area under the ROC curve. AUC (Area Under the Curve) is a more robust evaluation measure in anomaly detection as it considers the performance of the model across various thresholds. However, it has its limitations when applied to anomaly detection scenarios. AUC calculates the area under the entire ROC (Receiver Operating Characteristic) curve, whereas anomaly detection methods typically operate with fixed high sensitivity or high specificity requirement thresholds. Consequently, the area below these thresholds may not be relevant in evaluating the model's performance. To address this, using pAUC (partial AUC) that measures the area under the ROC above the sensitivity or specificity cut-off can provide more meaningful insights into the model's performance within the specific operating region (McClish, 1989).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad Specificity = \frac{TN}{FP + TN}$$

$$Precision = \frac{TP}{TP + FP} \qquad Sensitivity = Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \qquad GMean = \sqrt[2]{Sensitivity\, x\, Specificity}$$

Figure 2.1: Equation for Accuracy, Precision, F-measure, Sensitivity (Recall), Specificity, and GMean

In healthcare, anomaly detection utilizes data sources such as Electronic Health Records (EHR) (Estiri, Klann, and Murphy, 2019) and electrocardiogram (ECG) (Ngo and Veeravalli, 2015), with evaluation metrics such as sensitivity and specificity used to assess performance. Sensitivity measures the proportion of true positives (TP) out of all actual positives (true positives and false negatives), while specificity measures the proportion of true negatives (TN) out of all actual negatives (true negatives and false positives). Sensitivity indicates how well a test can correctly identify anomalies, while specificity measures the ability to correctly identify negative cases. However, sensitivity and specificity are usually inversely related, creating a trade-off where increasing one usually decreases the other. This trade-off can be challenging in situations where both high sensitivity and specificity are desired. Thus, sensitivity and specificity should be considered simultaneously, while also taking their interaction into account. Hence, a combination approach based on the geometric mean (GMean) has been utilized in previous studies evaluating anomaly detection methods (Dahmen and Cook, 2021; Maurya, Toshniwal, and Venkoparao, 2016). The equation for specificity, sensitivity, and GMean are shown in Figure 2.1.

In conclusion, selecting appropriate evaluation metrics is essential for accurately assessing the performance of anomaly detection techniques. Metrics such as accuracy, precision, recall, F-measure, ROC, and AUC provide valuable insights in various domains. However, the limitations of these metrics, particularly in imbalanced datasets or situations with different costs for false positives and false negatives, highlight the need for alternative approaches. Combining metrics, such as sensitivity and specificity using GMean, can offer a more balanced and informative evaluation criterion for anomaly detection methods.

2.4 Challenges in Outlier Detection

Outlier detection poses challenges in data, models, and evaluation metrics. Data-related considerations include understanding the dataset, identifying and resolving data quality issues, handling data heterogeneity, balancing data preprocessing, and considering data processing steps. Model-related considerations involve selecting appropriate techniques, adjusting model parameters, assessing parameter sensitivity to different anomaly types, and evaluating model performance. Result-related considerations include choosing suitable evaluation metrics, understanding the impact of false positives and false negatives, assessing the method's performance, and providing explanations for anomaly detection results. Addressing these challenges is crucial for developing effective anomaly detection solutions with accurate and interpretable results.

Current outlier detection methods have inherent limitations. The statistical approach relies on data's statistical properties to identify deviations from the expected distribution, offering simplicity and interpretability. However, it may fall short when dealing with complex patterns or multiple types of outliers. The nearest neighbor approach excels at detecting global, sparse, and local anomalies by analyzing proximity to neighboring instances. Yet, it struggles when anomalies have a sufficient number of close neighbors. The clustering approach groups similar observations into clusters, effectively detecting sparse and local anomalies. Nonetheless, it may encounter difficulties with densely-packed anomalies and datasets exhibiting complex shapes. The isolation approach, represented by the Isolation Forest, swiftly isolates anomalies through binary tree splits, making it proficient in detecting global outliers. However, it faces challenges in identifying local anomalies and accurately handling exponential data distributions.

In the main part of this report, a novel outlier detection technique named Outlier Reporting By Inference on Turfs (ORBIT) is introduced. ORBIT presents a novel and innovative approach to address the complexities of outlier detection, resulting in reliable and robust performance.

3 Outlier Reporting By Inference on Turfs

The organization of the chapter is as follows. The first section provides an introduction to ORBIT, a proposed solution for anomaly detection. It outlines the key concepts and objectives of ORBIT. The next section delves into the specifics of ORBIT. It covers the recommended input guidelines for ORBIT, including data formats and preprocessing steps. Additionally, it explores the different components of ORBIT and their roles in detecting anomalies. This section also discusses the reporting mechanism for outliers, explaining how the detected anomalies are reported and presented. The second focuses on the design considerations for each component of ORBIT. It highlights the rationale behind the design choices made and emphasizes the benefits they bring to the anomaly detection process. These considerations contribute to the overall performance, efficiency, and flexibility of ORBIT. Lastly, the third section examines the strengths of ORBIT by drawing on an understanding of fundamental anomaly detection approaches. It showcases how the design and methodology of ORBIT make it a more effective and efficient approach for detecting anomalies compared to other techniques. Through this structured approach, the chapter presents a comprehensive exploration of ORBIT, from its introduction to its design considerations and strengths.

3.1 Anomaly Detection Using ORBIT

The workflow of ORBIT, illustrated in Figure 3.1, comprises of three phases: input phase; turf inference and outlier detection phase; and output phase. The input phase involves assessing the characteristics of the data before feeding it into ORBIT. This step focuses on ensuring data quality, understanding feature distribution, and eliminating data heterogeneity. The turf inference and outlier detection phase comprises four main steps: reference point selection, turf inference, grouping and outlier detection. These components are designed to effectively detect outliers and contribute to the overall performance of ORBIT. The output phase involves presenting ORBIT's results to the user. This includes providing information on the confidence level of detected outlier instances and presenting visualizations of the identified outliers. By delivering clear and informative output, ORBIT enhances the interpretability and usability of the anomaly detection process. By providing a detailed overview of each aspect, from input to algorithm to output, this section offers a comprehensive understanding of ORBIT.

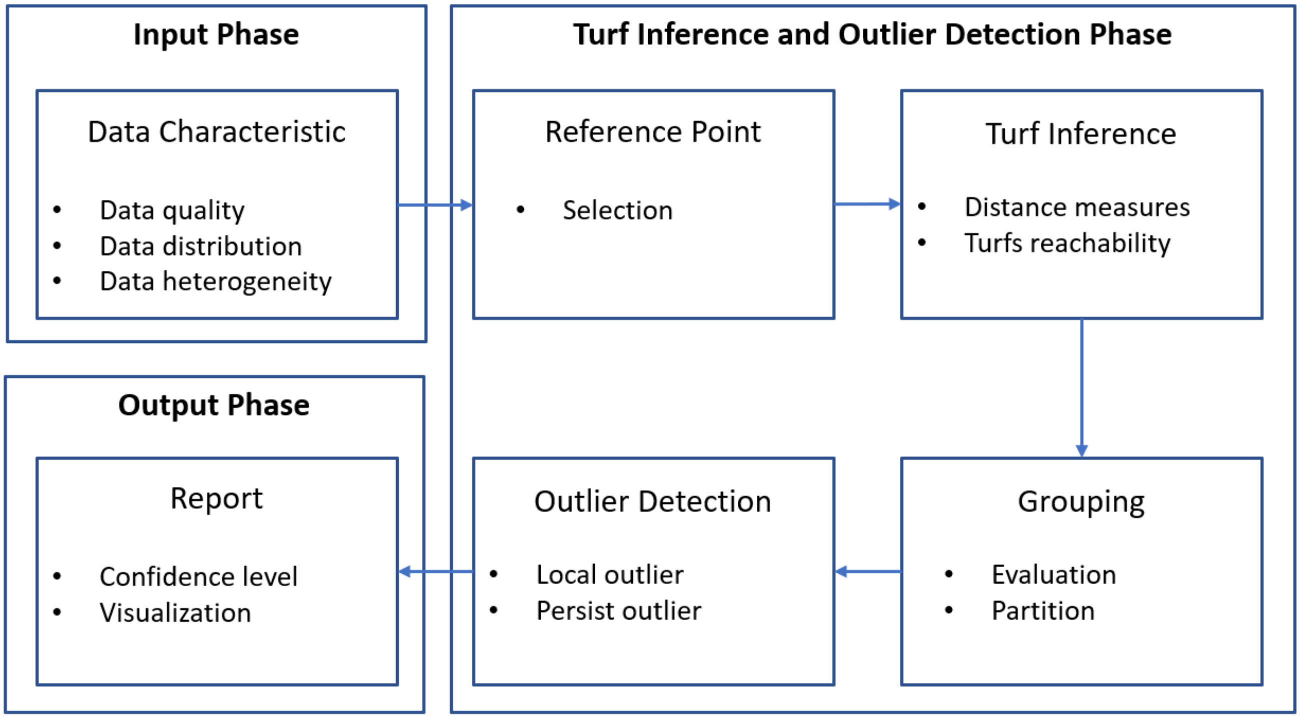


Figure 3.1: ORBIT workflow

3.1.1 Input Phase

The input part serves as a comprehensive guideline for understanding the characteristics of the data before running ORBIT. It encompasses three key aspects: data quality, data distribution, and data heterogeneity.

Data quality assessment focuses on identifying potential duplicate entries and missing data on an instance-wise level. Duplicate entries are removed to ensure they do not influence the percentage of outlier instances. In the case of missing data, two options exist: removing the data or imputing the missing values. In this report, due to time constraints, missing values in the dataset are handled by removing associated data instances from the analysis. This approach is chosen for its simplicity and efficiency, as it allows for a quick preprocessing step without the need for imputation methods. It is important to note that this approach assumes the missing values to be MCAR (missing completely at random) (Rubin, 1976). Therefore, discarding the data instances is equivalent to taking a random subsample, with the potential loss of important information if the missing values are actually MNAR (missing not at random) or MAR (missing at random). However, in the context of outlier analysis, this limitation is inconsequential, as the reduced dataset is not intended for any other analysis requiring the discarded data instances. Additionally, in this report, the implementation has been limited to handling datasets that contain exclusively numerical features. This decision was made to streamline the development process and prioritize addressing the specific challenges and characteristics associated with numerical data in anomaly detection.

Data distribution is a crucial consideration in anomaly detection. Some approaches make assumptions about the distribution of data to identify outliers, such as assuming a normal distribution in statistical approaches. However, these assumptions may not hold true for all datasets. In the case of ORBIT, it does not rely on any specific assumptions regarding the data distribution. This flexibility allows ORBIT to be applicable to a wide range of data distributions, making it a versatile anomaly detection solution that is not limited by assumptions about data distribution.

Data heterogeneity refers to situations where multiple features in the data exhibit different scaling or distributions. This can pose challenges in anomaly detection algorithms, particularly in cases where distance-based methods are used. One common approach to address data heterogeneity is through normalization, which aims to transform the data into a standardized range. In the case of ORBIT, the decision to employ min-max normalization is motivated by its ability to mitigate the impact of data heterogeneity. By scaling the features to a common range, min-max normalization ensures that the distance calculations between instances are not skewed by variations in feature scales. Previous literature reviews, such as the one conducted by Kandanaarachchi et al. (2020), have discussed the influence of normalization on nearest neighbor approaches. The experiment described in section 4.4.1 illustrates the impact of data heterogeneity on the performance of ORBIT, highlighting the importance of addressing this issue through appropriate normalization techniques.

By considering data quality, data distribution, and data heterogeneity, the input phase of ORBIT ensures that the data is appropriately prepared before processing it into the next phase. This comprehensive understanding of the data characteristics enhances the performance and reliability of the subsequent anomaly detection process.

3.1.2 Turf Inference and Outlier Detection Phase

The implementation of Turf Inference and Outlier Detection phase is outlined in Algorithm 1. It comprises four main steps, reference point selection, turf inference, grouping and outlier detection. Each step serving a specific purpose and collectively ensuring a systematic and organized approach to anomaly detection. These components play vital roles in the overall process, working together to accurately identify outliers within the dataset.

1. The Reference Point Selection step is responsible for selecting a set of reference point instances.

2. Turf Inference step is to estimate the typical distance between a reference point and its nearest neighbors, which defines the "turf" of that reference point. The turf of a reference point is determined by the region within a "reachability boundary" of t standard deviations from the median distance between the reference point and its set of nearest neighbors.

3. The Grouping step evaluates the selected reference points based on their turfs, discards those reference points with unusually large reachability boundary, and partitions the remaining reference

points into a specified number of distinct groups of similar sizes.

4. The Outlier Detection step identifies the outlier instances with respect to each group. These are termed "local outliers" of the respective groups. Then, it calculates the frequency at which each instance is classified as a local outlier. Those instances which persist as local outliers of many groups are reported as the detected outliers.

A synthetic dataset comprising of 2,000 instances with two-dimensional features was created specifically for explaining the working of ORBIT, as depicted in Figure 3.2a. Within the dataset, there are 20 outlier instances, accounting for approximately 1% of the total instances. Notably, each region along the x-axis and y-axis contains 10 outlier instances.

Algorithm 1 ORBIT algorithm

```
Input: N instances, $r$ number of reference points, $g$ number of groups, $k$ number of nearest neighbours, $t$ turf standard deviation threshold, $o$ outlier standard deviation threshold
Output: X outlier instances
Initialize an empty list R
Select $r \times g$ instances randomly from the dataset N and add them to list R
Build a k-d tree based on all instances in N
For each r_ in R:
    Get the top k nearest neighbors of r_ from the k-d tree
    Calculate the median (Median_r) and standard deviation (Sd_r) of the distances to the $k$ number of nearest neighbors
Calculate the median (Median_R) and standard deviation (Sd_R) of the Median_r values across all r_
Remove any r_ in R where Median_r $\geq$ Median_R + $t \times$ Sd_R
Partition the remaining r in R into g groups and store these groups in set G
Initialize an empty list X, with a length equal to the number of instances N, to track the classification of each instance as as a local outlier
For each g_ in G
    for each r_ in g_:
        Use the k-d tree to get instances in N within the turf boundary defined by Median_r + $t \times$ Sd_r of r_
    Identify instances in N that do not fall within any of the r_'s turfs and increase their count by 1 in X
Calculate the median (Median_X) and standard deviation (Sd_X) based on the count values in X
Return outlier instances x in X where x $\geq$ Median_X + $o \times$ Sd_X
```

The **Reference Point Selection step** implementation incorporates a random selection process to choose a set of reference point instances, which are then used for distance comparisons with all other instances. This approach provides a notable advantage in terms of computation efficiency. It reduces the time complexity from $O(N^2)$ to $O(N \times R)$ when computing nearest neighbors using basic nested loops, and from $O(N \log N)$ to $O(R \log N)$ when using a k-d tree for retrieving nearest neighbors. Here, N represents the number of instances in the dataset, and R represents the number of reference point instances. By adopting this approach, the computation cost is significantly reduced, resulting in faster outlier detection processes. The pseudocode for reference point selection can be found in Algorithm (1), specifically at line 2. The current selection algorithm randomly selects instances from the population while ensuring there are no duplicates. Figure 3.2b illustrates the randomly selected reference point instances, distinguished by a white circle. These selected instances represent 5% of the total instances in the dataset. This visual representation helps to illustrate the incorporation of reference points within ORBIT. The number of reference point instances (r) is a parameter associated with this component, and its guidelines and effects will be discussed in detail in a subsequent section.

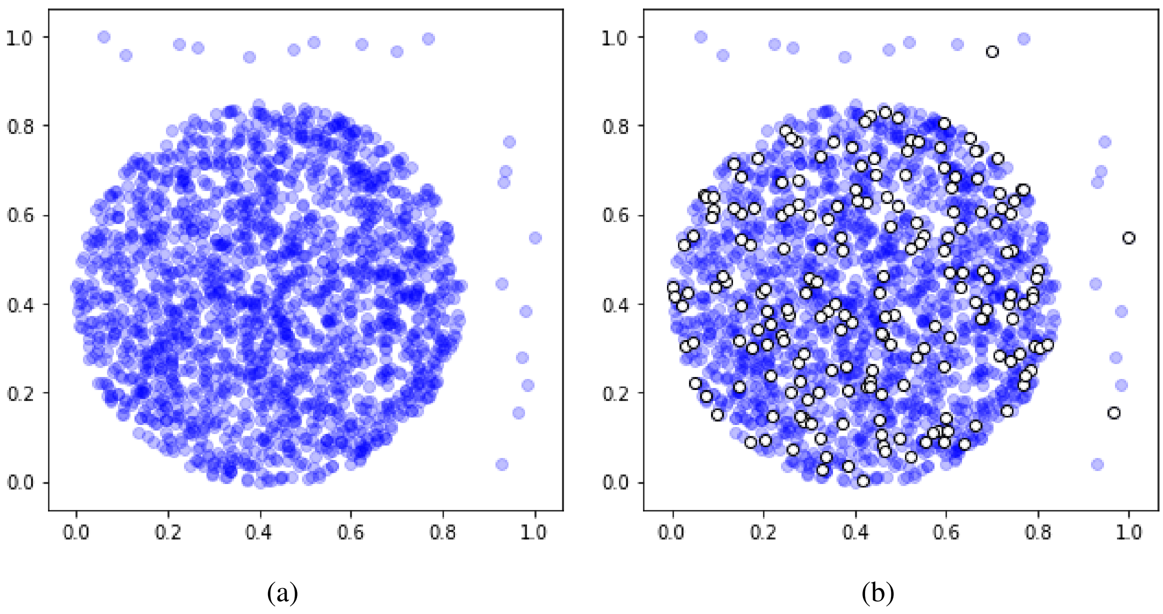


Figure 3.2: (a) Original instances dataset, and (b) Randomly selected reference point instances highlighted in black hollow circles.

The **Turf Inference step** encompasses two processes. Firstly, it involves computing the distances between the reference point instances and all other instances in the dataset. This computation allows for the assessment of the proximity between the reference points and the remaining data points. Secondly, the component determines the turf reachability boundary for each reference point. This boundary defines the spatial extent within which the influence of the reference points is considered during the outlier detection step.

To efficiently compute the distances between the reference points and other instances, a k-d tree (Friedman, Bentley, and Finkel, 1977) is constructed as outlined in Algorithm (1) at line 3. The next step involves determining the turf reachability boundary for each reference point. This is achieved by retrieving the top k nearest distances using the k-d tree function, which enables the calculation of the median and standard deviation of the top k nearest distances of a reference point, as depicted in lines 5 and 6 of the algorithm. Two parameters associated with this component are the number of nearest distances (k) and the threshold standard deviation value above the median (t), which defines the turf boundary. The guidelines and effects of these parameters are discussed in a later section. To provide a visual representation, Figure 3.3 illustrates the turf reachability boundaries for the selected reference point instances.

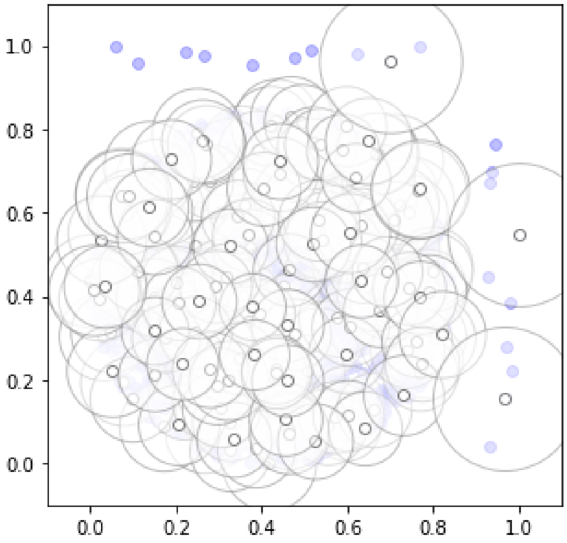
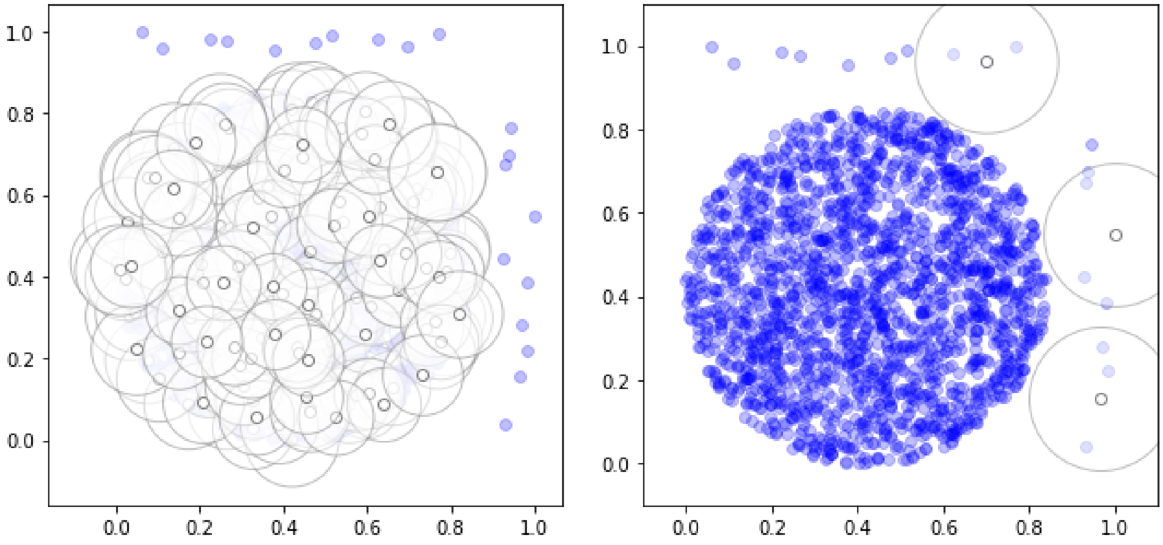


Figure 3.3: Turf reachability boundaries for the reference point instances determined by median and two standard deviations of the nearest 50 instances. Notice that the turf reachability boundaries of the three outliers on the right-hand side of the figure are significantly larger compared to the normal instances.

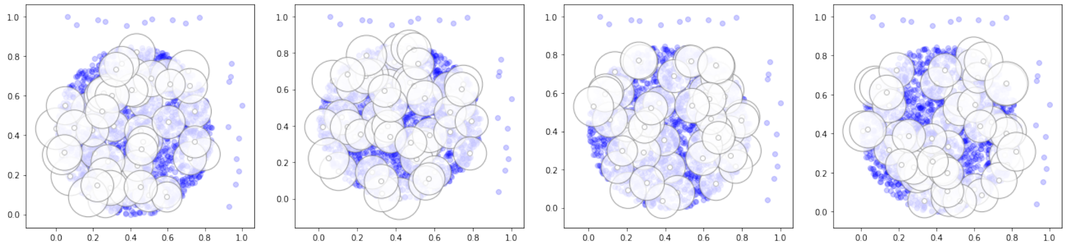
The **Grouping step** involves evaluating the selected reference point instances and partitioning a subset of them, those likely to be normal instances, into distinct groups. As the randomly selected reference point instances may themselves be outliers, it is crucial to assess their suitability for detecting nearby outlier instances. To accomplish this, the evaluation process calculates the median and standard deviation of all reference turf reachability boundaries, as shown in line 7 of the algorithm. Subsequently, reference points that deviate from the median by t standard deviations are removed, as depicted in line 8 of the algorithm.

After the evaluation, the remaining reference point instances are partitioned into a predetermined number of groups g. Each group is allocated a proportionate number of reference points through random assignment. Figure 3.4a showcases the reference points that remain after the evaluation process, while Figure 3.4b displays the reference points that have been removed. The remaining reference points are then randomly assigned to one of the groups. The resulting partition of reference points is depicted in Figure 3.5, highlighting a relatively even distribution across the four groups. The parameter g represents the number of groups and will be further discussed, including its guidelines and effects, in a subsequent section.



(a) (b)

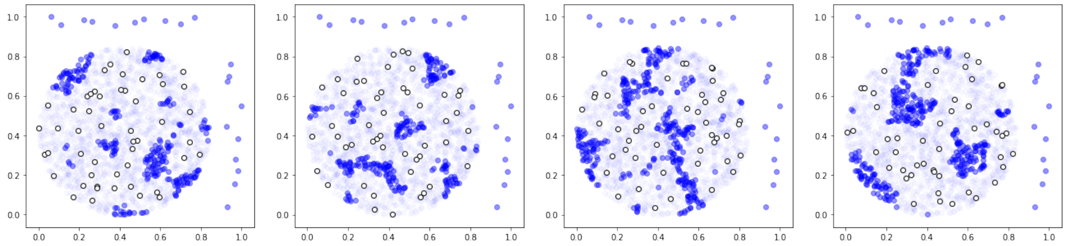
Figure 3.4: Results of reference point evaluation, where (a) displays the remaining reference points after evaluation, and (b) displays the removed reference points.



(a) (b) (c) (d)

Figure 3.5: Random splitting of reference points into four distinct groups.

The **Outlier Detection step** in ORBIT involves identifying outlier instances within each group (local outliers) and tracking the frequency of each instance being classified as a local outlier across all groups (persistent outliers). Algorithm (1) outlines the process for the outlier detection step (lines 10-16). Local outliers are instances that fall outside the reachability boundary of every reference point's turf within a group. To begin, a counter is initialized for each instance to track the number of times it is detected as a local outlier of a group (line 10). Within each group, the k-d tree function is utilized to identify instances within the turf boundary, which is determined by the median plus t standard deviation (line 13). Instances that do not fall within any reference point's turf in a group are considered local outliers, and their respective counters are incremented (line 14). Figure 3.6 visually presents the local outliers for the four groups, where opaque instances represent instances within the turf, and solid blue circles represent local outliers. The identification of persistent outliers involves counting the number of times each instance is classified as a local outlier across all groups. This is achieved by calculating the median and standard deviation of the local outlier counts over all instance (lines 15-16).



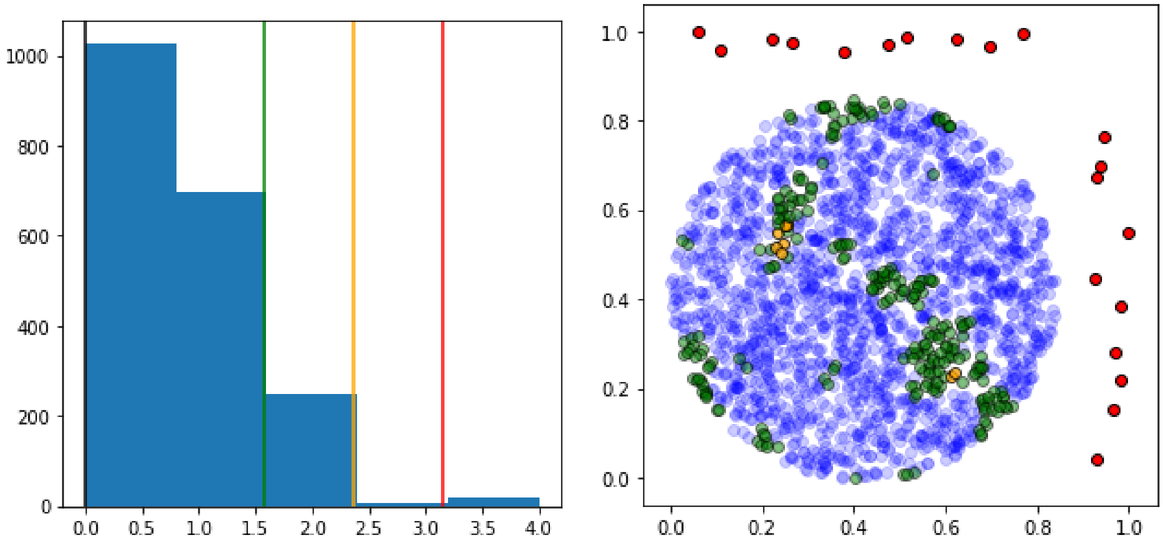
(a) (b) (c) (d)

Figure 3.6: Local outliers for each group are highlighted in a solid blue color, while reference points are represented by hollow circles.

To illustrate the significance of the local outlier count, Figure 3.7a presents a histogram that displays the frequency of local outlier counts. The x-axis represents the number of times an instance is identified as a local outlier of a group. The minimum value of zero indicates that the instance is not identified as a local outlier of any group, while the maximum value corresponds to the total number of groups, indicating that the instance is identified as a local outlier of all groups. The y-axis represents the number of instances falling under each local outlier count. The histogram also includes vertical lines indicating the median value (black line), plus 2 standard deviations (green line), plus 3 standard deviations (orange line), and plus 4 standard deviations (red line) away from the median. Additionally, Figure 3.7b displays a scatter plot illustrating the location of persistent outlier instances and their color-coding, similar to the vertical lines in the histogram. Outlier instances that are plus 2, plus 3, and plus 4 standard deviations away from the median are represented by the colors green, orange, and red, respectively. The parameter associated with this component is the threshold for the number of standard deviations away (o), which determines the extent to which instances are considered persistent outliers.

The motivation behind having tunable parameters in the ORBIT anomaly detection algorithm is to provide flexibility and adaptability to different datasets and application scenarios. Anomaly detection is a challenging task that varies greatly depending on the characteristics and nuances of the data being analyzed. Therefore, having tunable parameters allows users to customize the behavior of the algorithm to suit their specific requirements and achieve optimal results. ORBIT offers five tunable parameters, each influencing different aspects of the anomaly detection process. These parameters are as follows:

- The number of reference points r is a parameter that determines the quantity of instances sampled



(a)

(b)

Figure 3.7: The histogram plot (a) represents the persistent outlier count for each instance, displaying the frequency of outlier counts. The x-axis indicates the number of times an instance is detected as an outlier within a group, while the y-axis represents the number of instances falling under each outlier count. The scatter plot (b) visualizes the location of persistent outlier instances, using color-coding that corresponds to the vertical lines in the histogram.

as reference points. Increasing the value of this parameter results in a larger number of instances being selected as reference points, leading to an increase in pairwise distance calculations and longer computation time. The recommended range for this parameter is between 0.1% to 5% of the total instances, with a default value set at 0.25%.

- The number of groups g is a parameter that determines the quantity of groups used to evaluate whether an instance is normal or an outlier. A higher value increases the confidence level when identifying an instance as an outlier, but it also prolongs computation time due to additional pairwise distance calculations for the reference points within each group. The recommended range for this parameter is between 10 to 50, and the default value is set at 20 for datasets with more than a thousand instances, or 10 for datasets with fewer instances.

- The number of nearest neighbours k is a parameter used for inferring the boundary of the turf. A higher value results in a larger turf size, which can help reduce false positives. The default value for this parameter is set at 5% of the total number of instances.

- The turf reachability boundary standard deviation threshold t is a parameter that affects the size of the turf and the confidence level. A higher value increases the turf size and confidence level. The default value for this parameter is set at 2 standard deviations away from the median.

- The confidence level of the outlier detection standard deviation threshold o is a parameter that impacts the confidence level of outlier detection. A higher standard deviation threshold results in a higher confidence level for outlier detection. The default value for this parameter is set at 2 standard deviations away from the median.

In summary, tunable parameters in ORBIT enable customization, adaptability, and optimization for different datasets, requirements, and computation resources. These parameters provide users with the means to fine-tune the algorithm and achieve optimal anomaly detection performance in their specific context

3.1.3 Output Phase

The output of ORBIT consists of a comprehensive report that includes the identified outlier instances along with their corresponding confidence levels. Additionally, visualizations are provided to aid in understanding and interpreting the outliers. The confidence levels are determined based on the persistent outlier result, where instances that are plus 2, plus 3 and plus 4 standard deviations away from the median correspond to confidence levels of 96.46%, 98.62%, and 98.98% respectively. The confidence level for standard deviation was estimated using an empirical approach. The method involved generating 100 iterations on the 15 synthetic datasets (described in chapter 4.1.1) and determining the fractions of data instances that qualified as outliers to median + t standard deviations, where t = 2, 3, and 4. This process provided an empirical distribution, enabling the estimation of confidence levels associated with each standard deviation threshold.

To enhance the visual representation of the outlier instances, Figure 3.7b showcases a scatter plot where each instance is assigned a specific color code according to its confidence level. Normal instances are depicted in blue, while outlier instances with a 96.46% confidence level are represented by the color green. Outlier instances with a 98.62% confidence level are highlighted in orange, and those with a 98.98% confidence level are depicted in red. This color-coded visualization offers a clear distinction between different confidence levels, allowing for easy identification and analysis of outlier instances.

3.2 Design Consideration

In this section, the design considerations for each step of ORBIT are outlined. Table 3.1 presents a summary of these considerations, with italicized font indicating the final implementation for the algorithm and blue font indicating potential areas for future work. The following subsections will provide a description of each design consideration, along with an explanation of the rationale behind the chosen final implementation.

Step	Process	Algorithm Design Consideration			
Reference Point	Selection	R1: All instances	R2: random selection independently for each group	*R3: random selection dependently for all group*	R4: Perform clustering prior to random selection dependently
Turf Inference	Distance Measure	Euclidean distance numpy linalg.norm		*Neighbourhood search* *sklearn NearestNeighbors*	
		D1: *Single reference*	D2: Multiple random references	D3: multiple nearest references	
	Turf Reachability	*Median and standard deviation*			
Grouping	Evaluation	G1: Without reference filtering		*G2: With reference filtering*	
	Partition	*Random*		Clustering based on reference distance	
Outlier Detection	Local Outlier	*Binary where instance is within or outside the turfs*		Numerical of distance between instance and nearest reference	
	Persist Outlier	*O1: Empirical median and standard deviation*		O2: Chance or reliability index	O3: Sequential hypothesis test

Table 3.1: ORBIT design consideration

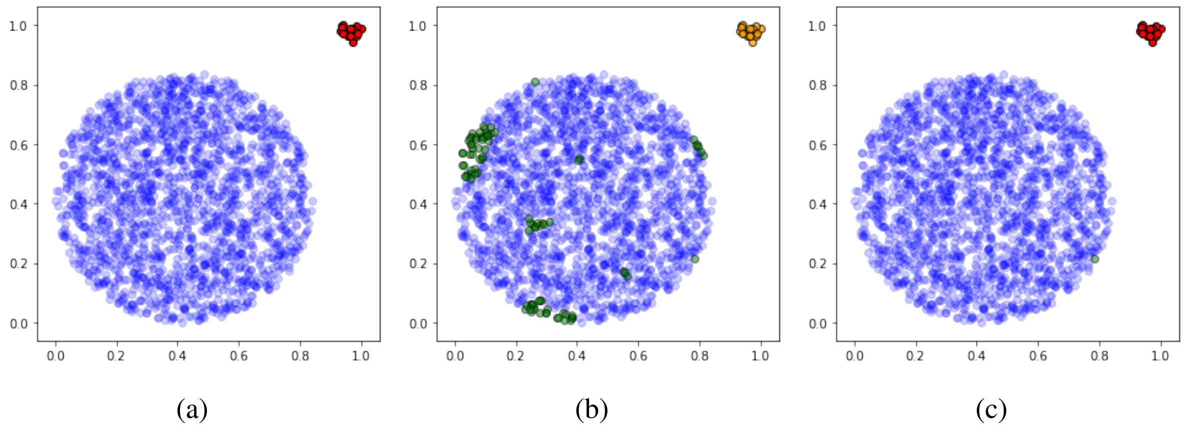
3.2.1 Reference Point Selection

The selection of reference point instances is a critical component in ORBIT and requires careful consideration to fulfill specific criteria. Two primary design considerations for reference point selection are computation time and memory efficiency, as well as representativeness. The selection algorithm should be fast, utilizing minimal memory resources, while ensuring that the chosen reference points are representative and evenly distributed throughout the dimensional space. Several potential implementations to address these considerations are outlined below.

Design R1: Using all instances as reference points. The first implementation involves using all instances as reference points. This approach has the advantage of eliminating bias by considering all instances in proximity measurements. However, it shares limitations with the local outlier factor (LOF) nearest neighbor implementation, resulting in a high computation complexity of $O(N^2)$.

Design R2: Random selection of instances as reference points independently for each group. This implementation entails independently selecting a set of instances as reference points for each group. One advantage of this approach is its computation efficiency, with a complexity of $O(N \times R)$, where N is the number of instances and R is the number of reference points. However, there is a potential risk if a group happens to include an outlier instance as a reference point. Additionally, groups may contain duplicated reference points, which may result in reduced coverage of normal instances later on.

Design R3: Random selection of instances as reference points for all groups. This is the current implementation of ORBIT, which involves selecting a set of instances as reference points for all groups. This approach offers the advantage of avoiding duplicate instance selection. Furthermore, as all reference points are evaluated in subsequent steps of the algorithm using the entire population, the



(a) (b) (c)

Figure 3.8: Outlier detection results for different reference point selection implementations: (a) R1 using all instances (50.87 seconds), (b) R2 with independent random selection for each group (9.44 seconds), and (c) R3 with dependent random selection for all groups (10.01 seconds).

likelihood of selecting an outlier instance as a reference point is further diminished.

Design R4: Performing clustering before random selection of instances as reference points. A potential future implementation of ORBIT involves performing clustering and selecting a set of instances as reference points from each cluster, taking into account the cluster size. This approach can help prevent the misselection of likely outlier instances as reference points from the smallest clusters, as well as ensuring improved coverage of normal instances by the reference points.

To demonstrate the various implementations of reference point selection, Figure 3.8 showcases the results of ORBIT's outlier detection on global sparse outlier data. The dataset comprises 2000 instances with 2-dimensional features, of which 20 instances (1%) are outlier instances. The algorithm was executed with the following parameter settings: 10 reference points, 10 groups, 50 nearest neighbours, and a turf boundary of 2 standard deviations. The computation time for R1, R2, and R3 implementations were 50.87, 9.44, and 10.01 seconds, respectively. While R1 and R3 yielded similar outlier detection results, R1 exhibited a computation complexity of $O(N^2)$, whereas R3 had a computation complexity of $O(N \times R)$. The outlier detection performance of R2 was slightly lower than that of R3, as evidenced by lower confidence in detecting outlier instances (highlighted in orange) and some false positives (highlighted in green). This disparity can be attributed to the subsequent steps of ORBIT, where the reference point evaluation only compares instances within the same group for R2, which has limited statistical capability in removing outlier reference points.

3.2.2 Turf Inference

There are two primary implementations in the distance and turf component: distance measure implementation, and turf reachability implementation.

Pairwise distance calculation is a crucial step in ORBIT for measuring distances between reference points and instances. Two Python libraries, numpy's linalg.norm and sklearn's NearestNeighbors, were explored. The numpy approach computes pairwise Euclidean distances. These distances are stored in a dictionary and sorted using Timsort, with a worst-case complexity of $O(N \log N)$ (Auger et al., 2018), to find the k nearest neighbors. To handle larger datasets efficiently, only the k nearest distances are stored and continuously updated as additional distances are calculated. On the other hand, the current ORBIT implementation utilizes sklearn's KDTree for nearest neighbor searches, resulting in a significant reduction in computation time. Experiment results on synthetic data (described in chapter 4.1.1) showed comparable evaluation scores, with average runtimes of 379.98 seconds for numpy and 16.86 seconds for sklearn.

To enhance turf reachability away from global outlier instances, an additional design consideration for the distance measure is to compare each instance with multiple reference points. The rationale behind this approach is based on the possibility of the selected reference point being an outlier instance. By summing the distance of an instance to multiple reference points, it might reduce the risk of an outlier being evaluated as a normal instance. Consequently, the identified outlier instances are those that lie outside the overlapping turfs of multiple reference points.

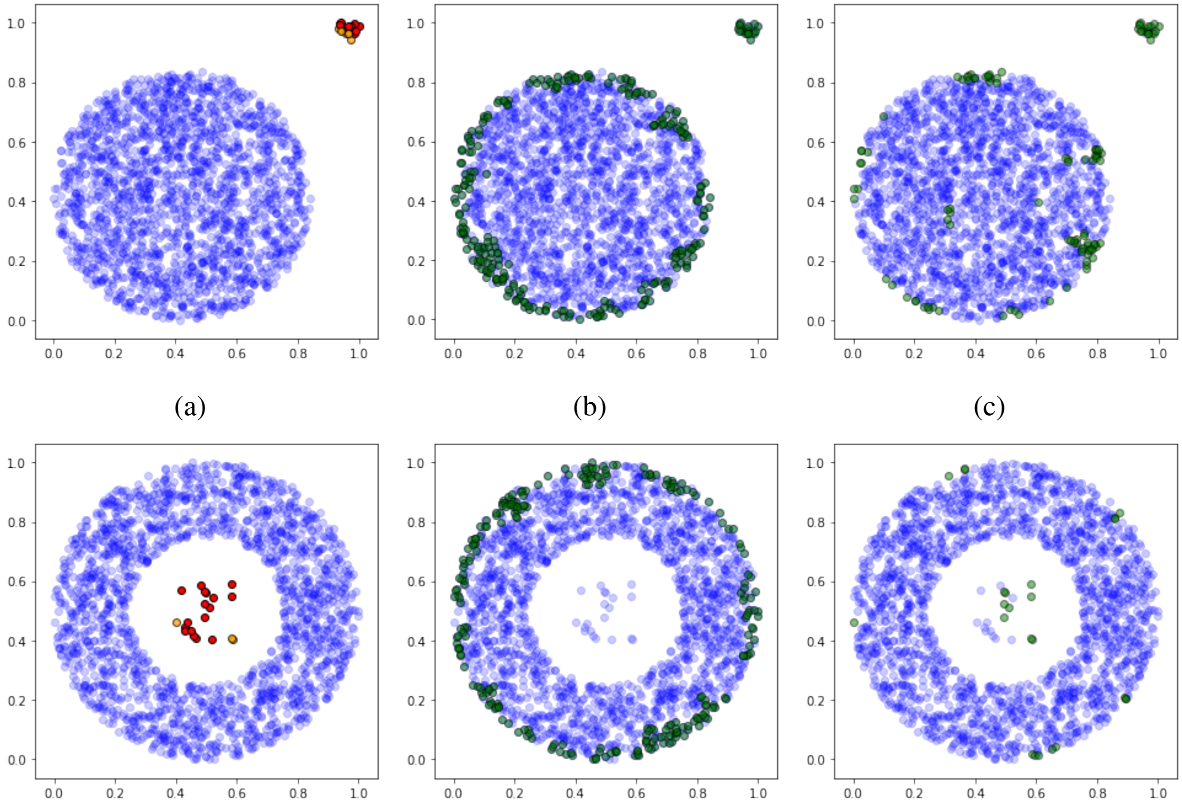
Figure 3.9 displays the results of three distance measure implementations: single reference (D1), multiple random references (D2), and multiple nearest references (D3) on two datasets of global outliers (a, b, c) and local outliers (d, e, f). The D2 implementation has several potential issues because the multiple reference points are chosen randomly, which may result in instances at the edge being incorrectly identified as outliers and failing to identify the local outliers. The D3 implementation improves upon D2 by selecting other reference points based on instances that are closest to the reference point of interest, resulting in a turf reachability boundary closer to the reference point of interest. However, both D2 and D3 implementations still exhibit poor performance in detecting local outliers. The final implementation of ORBIT's distance measure is based on comparing instances to a single reference point.

In the implementation of **turf reachability**, the algorithm utilizes the median of the k nearest distances, complemented by a user-defined standard deviation with a default value of 2. The choice to employ the median, rather than the mean, is motivated by the fact that the mean can be susceptible to the influence of outlier instances within the nearest neighbours. In contrast, the median offers a more robust and resistant measure against such outliers.

3.2.3 Grouping

The grouping step of ORBIT involves two main components: the evaluation of reference points and the division of the remaining reference points into multiple groups.

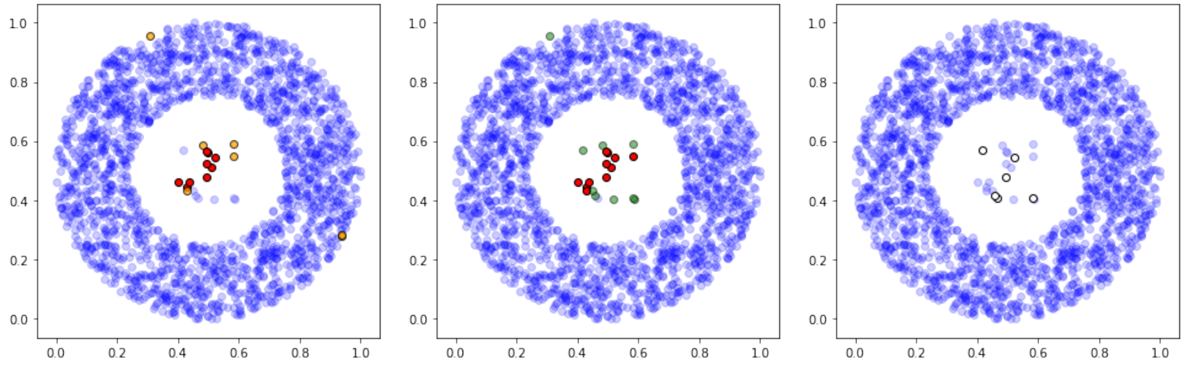
In the **evaluation** of reference points, two design considerations have been explored: The first design (G1) utilizes all of the selected reference points, which may include instances that could potentially



(a) (b) (c)

(d) (e) (f)

Figure 3.9: Comparison of distance measure implementations on global outlier datasets (a, b, and c) and local outlier datasets (d, e, and f): (a, d) Single reference point, (b, e) Multiple random references, and (c, f) Multiple nearest references.



(a) (b) (c)

Figure 3.10: Comparison of outlier detection results between two designs: (a) utilizing all selected reference points (G1) and (b) evaluating reference points based on their turf reachability boundaries (G2). Figure (c) visualizes the reference points that have been removed, representing the points used in G1 but not in G2.

be outliers. While, the second design (G2) compares the turf reachability boundaries of all reference points and removes any reference points that exceed the median plus *n* standard deviation of these turf reachability boundaries. The purpose of this removal is to eliminate any potential outlier instances that are selected as reference points. Figure 3.10 demonstrates the difference in outlier detection results between the two designs: (a) using all selected reference points (G1) and (b) evaluating reference points based on their turf reachability boundaries (G2). Furthermore, Figure 3.10c provides a visualization of the reference points that have been removed, it is the reference points used in G1 but not in G2.

The current approach used in ORBIT for assigning reference points into multiple groups is based on random distribution. However, in the G2 design, the evaluation of reference points may lead to a different number of reference points in each group. Figure 3.4 illustrates this variation, where three reference points are removed out of the initial 100 reference points after the evaluation in the G2 design. The remaining 97 reference points are divided into four groups, with one group containing 25 reference points and the other three groups each consisting of 24 reference points. While random distribution can be effective for assigning reference points into multiple groups, it has a potential drawback. In some cases, instances of reference points within a single group may be closely located to each other. This proximity of reference points can impact the accuracy of outlier detection. To address this limitation and improve the distribution of reference points within each group, a future implementation could incorporate clustering techniques. By clustering the reference points and considering the distance between them, the division of reference points into multiple groups can be better balanced. This approach would result in a more diverse distribution and coverage of reference points within each group, ultimately enhancing the performance of outlier detection.

3.2.4 Outlier Detection

The outlier detection component consists of two main components: local outlier and persistent outlier.

The current implementation of the **local outlier** component in ORBIT uses a binary approach to determine if an instance belongs to the turfs of reference points within a group. However, there is room for improvement by considering the distance between an instance and its nearest reference point. By taking into account this distance, the outlier detection process can be refined. This enhanced approach would assign a higher penalty to instances that are located farther away from the reference point within the same group. As a result, the outlier detection results might be more precise and accurate, potentially improving the ability to identify outliers effectively.

In ORBIT, the current approach for identifying **persistent outliers** involves counting the number of times an instance is labeled as a local outlier across different groups (O1). However, there are alternative methods that can be explored to improve performance. One alternative is the use of a reliability index (O2). this method calculates the probability of falsely detecting an instance as an outlier. By assigning a higher weight to instances that consistently appear as outliers, the reliability index reduces the chances of false detections and enhances accuracy. Another alternative is sequential hypothesis testing (O3). This method performs statistical tests after each group to determine if there is sufficient evidence to conclude or reject that data instance as an outlier based on the $r < n$ groups already analyzed. This approach enables informed decisions based on the results obtained so far, potentially reducing the number of evaluations required and improving efficiency. Exploring these alternative methods can enhance the efficiency and accuracy of the persistent outlier component in ORBIT. By considering these options, the algorithm can become more effective in identifying outliers and improve overall performance.

3.3 Strength of ORBIT

ORBIT demonstrates several strengths that stem from its components and steps, which aim to improve computation efficiency, facilitate understanding, and enhance outlier detection accuracy.

One of the merits of ORBIT is its assumption that datasets mainly comprise a large number of normal instances with only a few outliers. This characteristic allows ORBIT to randomly select a small sample of "references", which is more likely to consist of minimal or no outliers. The reference selection component in ORBIT actively embraces this principle, enabling the algorithm to utilize the representative characteristics of the selected references and improve the accuracy of outlier detection.

Another strength of ORBIT is its assumption that most instances are normal and that normal instances are closer to other normal instances compared to outliers. This implies that the nearest neighbors of the normal reference instances are more likely to be normal rather than outliers. This characteristics allows ORBIT to utilize these neighbours and their distances from the reference instance to determine the typical distances of normal instances from this reference. The "turf inference" component in ORBIT embraces this principle by defining the turf reachability boundary for a normal region as the radius of the median plus *n* standard deviations of these neighbor distances.

The grouping component of ORBIT serves the purpose of randomly dividing the set of references into multiple groups. This division ensures that each group consists of references that are sufficiently dispersed. Consequently, the combined turfs covered by these references have a higher probability of encompassing most of the normal regions. By incorporating an adequate number of reference groups, ORBIT ensures comprehensive coverage of a majority of normal regions, thereby enhancing its outlier detection performance and robustness.

In scenarios where a normal instance falls outside the turf of multiple groups, the persistent outliers component of ORBIT plays a crucial role. This component identifies instances that consistently fall outside the turfs of most (if not all) groups and treats them as outliers. To achieve this, the distribution of the number of times an instance falls outside the turfs of a group is analyzed. This distribution is predominantly composed of normal instances, in line with the underlying assumption that a majority of instances are normal. Conversely, outliers are located at the extreme right end of the distribution. To establish a threshold for identifying outliers, ORBIT applies a practical rule of thumb. This involves setting the threshold as the median plus *n* standard deviations. This approach allows ORBIT to effectively identify instances exceeding the threshold as outliers, further improving the accuracy of outlier detection.

Overall, the components and steps of ORBIT work synergistically to improve computation efficiency, simplify understanding, and enhance outlier detection accuracy, making it a robust approach for anomaly detection tasks.

4 Empirical Evaluation

This chapter presents the results of four comprehensive sets of experiments conducted to evaluate the effectiveness of ORBIT. The first experiment is aimed at assessing ORBIT's performance across various types of anomaly characteristics. It provides a comparative analysis with other well-established anomaly detection methods, allowing for a comprehensive evaluation of ORBIT's effectiveness in different scenarios. The second experiment focuses on evaluating the impact of different parameter settings on ORBIT's performance. This analysis provides valuable insights into the algorithm's behavior and enables the identification of optimal settings for different datasets. In the third experiment, the algorithm's performance is evaluated on datasets with heterogeneous characteristics, diverse topologies, high dimensions, and large amounts of instances. This evaluation aims to assess ORBIT's robustness and adaptability in challenging real-world scenarios. Lastly, the fourth experiment evaluates ORBIT's performance on various benchmark datasets widely used in the field of anomaly detection. This comparative analysis allows for a comprehensive assessment of ORBIT's performance against several better-known and more recent anomaly detection approaches. These better-known anomaly detection approaches are Median Standard Deviation statistical approach (Dave and Varma, 2014), K-means clustering (Breunig et al., 2000), Local Outlier Factor (LOF) nearest neighbor (Chawla and Gionis, 2013), and Isolation Forest (IF) (Liu, Ting, and Zhou, 2008). While the more recent anomaly detection approaches are ABOD (Kriegel, Schubert, and Zimek, 2008), COPOD (Li et al., 2020), ECOD (Li et al., 2022), HBOS (Goldstein and Dengel, 2012), and ROD (Almardeny, Boujnah, and Cleary, 2020). By conducting these four sets of experiments, this chapter provides a comprehensive evaluation of ORBIT's performance, efficiency, robustness, and comparative effectiveness. The results obtained from these experiments contribute to a deeper understanding of ORBIT's capabilities and its potential as a useful anomaly detection solution.

4.1 Experiment Design

4.1.1 Evaluation Data

To evaluate the effectiveness of ORBIT, a comprehensive set of experiments was conducted using both synthetic and benchmark datasets. The inclusion of these datasets allows for a comprehensive assessment of ORBIT's performance across different data types and characteristics. The synthetic datasets were designed to simulate various types of anomalies, including global outliers, local outliers, and anomalies with different degrees of density. These datasets provide controlled environments for evaluating ORBIT's ability to accurately detect and classify anomalies based on their characteristics. In addition to the synthetic datasets, benchmark datasets widely used in the field of anomaly detection were also employed. These datasets represent real-world scenarios and have been extensively studied by the research community. By evaluating ORBIT's performance on these benchmark datasets, a meaningful comparison with other anomaly detection approaches can be established.

Synthetic Datasets

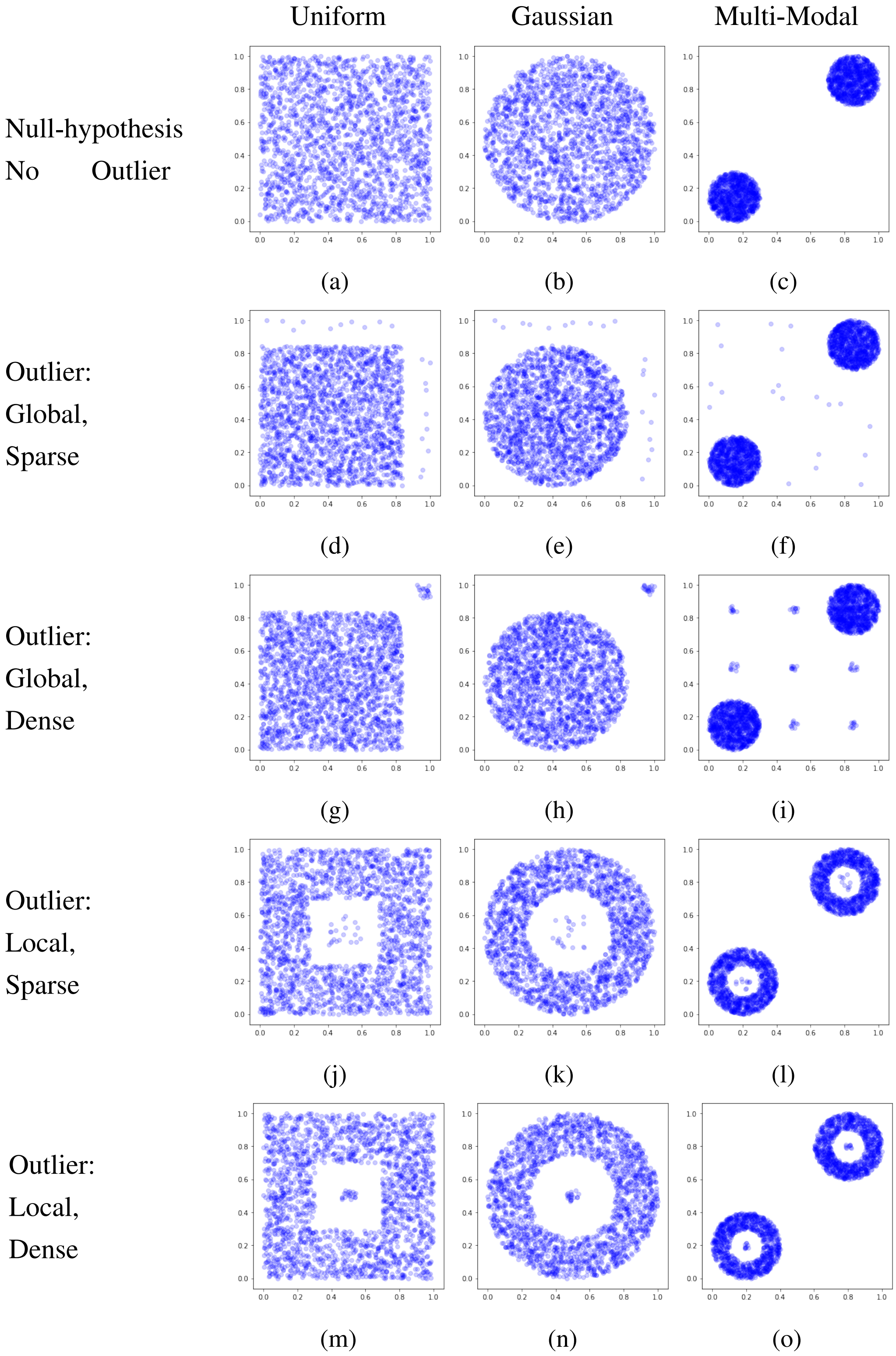
The synthetic datasets generated for evaluating the performance of various anomaly detection methods encompass different types of anomalies. In order to comprehensively evaluate the proposed solution, the initial experiments focus on generating synthetic datasets that test specific data characteristics and different types of anomalies. These characteristics and anomalies include:

- Data distributions: The datasets cover various data distributions, such as uniform distribution, Gaussian distribution, and multi-modal distribution.
- Null hypothesis: Datasets are created with no anomalies present.
- Outlier location: Anomalies are generated as either global outliers (outside the normal instances) or local outliers (surrounded by normal instances).
- Outlier spread: Anomalies exhibit either sparse spread (spread out) or dense spread (closely located).

Figure 4.1 illustrates the 15 sets of synthetic data created to assess the performance of anomaly detection. Each dataset consists of 2,000 instances with 2 features for visualization purposes. Among the synthetic datasets that contain anomalies, 20 instances out of the 2,000 are anomalies, representing 1% of the data. The multi-modal global dense synthetic dataset is an exception, as it comprises 2,060 instances, of which 70 (3.3%) are anomaly instances. The design of the global dense synthetic dataset incorporates three levels of difficulty for outlier instances. The first level includes outliers located at the edges, far away from the blobs of normal instances. The second level involves outliers positioned in the middle, closer to the blobs of normal instances. The third level encompasses outliers located between the two blobs, within the inner part of the dataset.

Benchmark Data

The benchmark dataset selection process adhered to several criteria. Firstly, the datasets had to be widely used in the field of anomaly detection. Secondly, they needed to consist of tabular point data where each instance contained multiple features. Thirdly, the datasets had to provide ground truth labels and should not contain categorical features or missing values. Lastly, the percentage of outlier instances in the dataset was required to be less than 10% of the total number of instances. Table 4.1



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 4.1: Synthetic datasets

showcases the selected benchmark datasets from various domains, listed in ascending order based on the number of instances they contain. The benchmark data used in this study is sourced from *ODDS Outlier Detection Data Sets - odds.cs.stonybrook.edu*.

No	Name	# Instance	# Dimension	# Outlier (%)
01	Wine	129	13	10 (7.70%)
02	Lympho	148	18	6 (4.10%)
03	Glass	214	9	9 (4.20%)
04	WBC	278	30	21 (5.60%)
05	Vowels	1,456	12	50 (3.40%)
06	Letter	1,600	32	100 (6.25%)
07	Musk	3,062	166	97 (3.20%)
08	Speech	3,686	400	61 (1.65%)
09	Thyroid	3,772	6	93 (2.50%)
10	Optdigits	5,216	64	150 (3.00%)
11	Satimage-2	5,803	36	71 (1.20%)
12	Pendigits	6,870	16	156 (2.27%)
13	Annthyroid	7,200	6	534 (7.42%)
14	Mnist	7,603	100	700 (9.20%)
15	Mammography	11,183	6	260 (2.32%)
16	Shuttle	49,097	9	3,511 (7.00%)
17	Smtp (KDDCUP99)	95,156	3	30 (0.03%)
18	ForestCover	286,048	10	2,747 (0.90%)
19	Http (KDDCUP99)	567,479	3	2,211 (0.40%)

Table 4.1: Summary of Benchmark Data

4.1.2 Other Anomaly Detection Approaches

This report evaluates several other anomaly detection approaches, which are evaluated alongside the ORBIT to compare their performance.

- The statistical approach based on Median Standard Deviation (Dave and Varma, 2014) utilized the Python numpy library. This approach calculates the median and standard deviation for each feature dimension independently. Instances that are located a certain number of standard deviations away from the median of any single feature are identified as outliers.
- Clustering approach based on distance from cluster centroid (Chandola, Banerjee, and Kumar,

2009): The implementation of this approach utilized the Python sklearn KMeans library. It employs the KMeans clustering technique based on Lloyd's algorithm (Lloyd, 1982). After clustering is completed, the euclidean distances between each instance and its centroid are computed. Within each cluster, the median and standard deviation of these distances are calculated. Instances that deviate a certain number of standard deviations from the median distance within each cluster are identified as outliers.

- Clustering approach based on cluster size (Chawla and Gionis, 2013): The implementation of this approach also utilized the Python sklearn KMeans library. It involves performing KMeans clustering with a large value of k. Outlier instances are identified as those belonging to smaller clusters containing less than a certain percentage of the total instance count.

- Nearest neighbor approach based on Local Outlier Factor (LOF) (Breunig et al., 2000): The implementation of this approach utilized the Python sklearn LOF library. The LOF implementation involves several key steps to identify and quantify the outlierness of data points. Firstly, the algorithm performs a nearest neighbor search for each data point. By finding the k nearest neighbors of each point, where k is a predetermined value based on dataset characteristics, the algorithm captures the local structure and relationship between data points. Once the nearest neighbors are determined, the algorithm calculates the local reachability density (LRD) for each data point. LRD measures the density of a data point's local neighborhood by considering the distances to its k nearest neighbors. The average inverse reachability distance from the data point to its neighbors is used to compute the LRD. The reachability distance can be either the actual distance or the distance to one of the neighbors, depending on the specific implementation. Next, the algorithm calculates the local outlier factor (LOF) for each data point. LOF compares the average LRD of a data point's k nearest neighbors to its own LRD. If the average LRD of the neighbors is significantly higher than the LRD of the data point itself, the LOF value will be greater than 1, indicating that the data point is denser than its neighbors and is more likely an inlier. Conversely, if the LOF value is significantly lower than 1, it suggests that the data point is less dense than its neighbors and is more likely an outlier. Finally, the algorithm applies a threshold to the LOF values to identify outliers. Data points with LOF values exceeding the threshold are considered outliers. The choice of the threshold can be determined using statistical methods or domain knowledge.

- Isolation approach based on Isolation Forest (IF) (Liu, Ting, and Zhou, 2008): The implementation of this approach utilized the Python sklearn IsolationForest library. The IF algorithm works by randomly selecting a feature and a split value within the range of that feature for the dataset. Using the selected feature and split value, the algorithm partitions the data into two subsets: one subset containing data points with values below the split value and another subset containing points with values above the split value. This process of feature selection and data partitioning is recursively applied to each subset until the data points are completely isolated or a maximum depth is reached. To score outliers, the algorithm records the path length from the root of each isolation tree to reach a data point. Outliers are expected to have shorter path lengths because they are easier to isolate, while inliers require more splits to isolate them. The path lengths are averaged across all trees to obtain an anomaly score for each data point. Finally, the algorithm identifies outliers by applying a specified threshold to the anomaly scores. Data points with scores above the threshold are considered outliers.

- Other outlier detection approaches utilized in this study are ABOD (Kriegel, Schubert, and Zimek, 2008), COPOD (Li et al., 2020), ECOD (Li et al., 2022), HBOS (Goldstein and Dengel, 2012), and ROD (Almardeny, Boujnah, and Cleary, 2020). These implementations make use of the Python PyOD library, as published by Zhao, Nasrullah, and Li (2019). The experiments in this report are conducted using the default parameter settings provided by the PyOD implementation.

4.1.3 Parameter Settings

There are five parameter settings in ORBIT. The settings used in the experiments below are as follows. The number of reference points for each group is set at 1.25% of the total number of instances, while the number of groups is fixed at 20. To determine the turf size, the algorithm considers 2.5% of the total number of instances as the nearest distances. A standard deviation threshold is used to establish the turf reachability boundary, and it is set at 2. Additionally, a standard deviation threshold of 2 is employed to determine the confidence level of outliers.

Regarding the parameter settings for the other methods, each approach utilizes different configurations. In the Statistical Median Standard Deviation approach, the standard deviation threshold away from the median is used, and it has a default value of 2. For the KMeans approach based on distance from centroid, the parameters include the number of clusters (k) and the standard deviation threshold, which have default values of k = 2 and standard deviation threshold = 2. In the KMeans approach based on cluster size, the number of clusters (k) and the percentage threshold of cluster size to the total number of instances are considered, with default values of k = 10 and percentage threshold = 5%. For example, if a dataset contains 1,000 instances, the outlier instances are defined as clusters consisting of 50 or fewer instances. For the Local Outlier Factor (LOF) approach, the parameter setting involves determining the number of nearest neighbors to compare the density with, and it has a default value of 5% of total number of instances. In the case of the Isolation Forest (IF) approach, multiple parameters are used, including the number of sub-samplings, the number of features in a tree, the number of trees to build, and the containment size. The default values for these parameters are 256, 10, 100, and 5% of the total number of instances, respectively. COPOD, ECOD, and ROD each have a single parameter setting called contamination, which defines the proportion of outliers in the dataset. This parameter is used to set the threshold on the decision function, with the default value being 0.1. ABOD has two parameters, namely number of neighbors and contamination. The number of neighbors parameter determines the default value for k-neighbors queries, which is set to 5, while the default contamination value is 0.1. For HBOS, there are four parameters, namely number of bins for the histogram, regularizer for preventing overflow, flexibility when handling samples falling outside the bins, and contamination. The default values for these parameters are 10, 0.1, 0.5, and 0.1, respectively.

4.1.4 Evaluation Criteria

In order to evaluate the performance of various anomaly detection implementations, it is crucial to employ evaluation metrics that are reliable and robust. Section 2.3 provided a comprehensive overview of different evaluation metrics commonly used in the field of anomaly detection. Among these metrics, sensitivity and specificity have proven to be particularly valuable for assessing implementation effectiveness. Sensitivity measures the proportion of correctly identified outlier instances. A sensitivity score of 100% indicates that every outlier instance is accurately recognized as an outlier. On the other hand, specificity measures the proportion of correctly identified normal instances. A specificity score of 100% implies that no instances are falsely classified as outliers. The decision to employ sensitivity and specificity is based on several rationales. Firstly, these evaluation criteria can be consistently applied across datasets of different sizes. Additionally, they remain unaffected by changes in prevalence between different classes of data instances. To thoroughly assess the performance of the current implementation, it is important to consider both measures together. In addition to sensitivity and specificity, a combination approach known as the geometric mean (GMean) is also utilized. The GMean of sensitivity and specificity is a robust metric for evaluating outlier detection performance, given the expected highly imbalanced class distributions. By focusing on overall outlier detection performance and providing a single interpretable value, the geometric mean offers a balanced assessment of both detecting true outliers and correctly identifying non-outliers, making it useful for comparing outlier detection algorithms across different scenarios.

4.1.5 Experiment Environment

The experiment environment is of utmost importance when conducting reliable and valid experiments. It encompasses the hardware, software, and other necessary conditions for the experiments. In this study, all experiments were implemented using Python version 3.7 and executed on an Azure Linux virtual machine. The virtual machine had a configuration of VM Standard_D4ads_v5, equipped with a 4 CPU 2.40GHz Intel i5 processor and 16GB of memory. The operating system of the virtual machine was Red Hat Enterprise Linux Server release 7.6.

To ensure the data's integrity and consistency, several data preparation steps were undertaken. These steps involved cleaning the data by removing instances that contained empty or missing values. Additionally, data heterogeneity was addressed by performing Minimum and Maximum normalization, which standardized the data across its range. These preprocessing steps were essential to enhance the quality and suitability of the data for the subsequent analysis and anomaly detection tasks.

4.2 Experiment: Synthetic Datasets

In this section, the results and discussion for ORBIT and five other anomaly detection methods described in the previous section are presented. Table 4.2 provides a summary for the GMean and computation time data for the 15 synthetic datasets.

No	ORBIT (2 SD)		Median Standard Deviation		Local Outlier Factor		Kmeans - distance to centroid		Kmeans - cluster size		Isolation Forest	
	GMean	Time	GMean	Time	GMean	Time	GMean	Time	GMean	Time	GMean	Time
a	0.958	1.499	**1.000**	0.005	**1.000**	0.018	0.999	3.135	**1.000**	8.004	0.995	0.257
b	0.988	1.497	0.999	0.007	**1.000**	0.024	0.987	3.206	**1.000**	7.819	0.995	0.271
c	0.953	1.436	**1.000**	0.007	**1.000**	0.023	**1.000**	0.612	**1.000**	3.108	0.995	0.267
d	0.965	1.528	**1.000**	0.008	**1.000**	0.024	0.805	3.288	0.000	6.187	0.893	0.289
e	0.966	1.526	**1.000**	0.008	**1.000**	0.024	0.911	4.356	0.000	7.392	**1.000**	0.270
f	**1.000**	1.534	0.000	0.007	**1.000**	0.024	**1.000**	0.903	0.949	3.049	**1.000**	0.259
g	**1.000**	1.531	**1.000**	0.008	**1.000**	0.024	**1.000**	2.849	**1.000**	6.025	**1.000**	0.256
h	**1.000**	1.529	**1.000**	0.008	**1.000**	0.025	0.993	6.448	**1.000**	8.181	**1.000**	0.276
i	**1.000**	1.542	0.000	0.005	**1.000**	0.025	**1.000**	1.177	0.756	3.083	0.548	0.249
j	**0.977**	1.541	0.000	0.009	0.866	0.024	0.000	3.073	0.000	7.481	0.000	0.259
k	0.945	1.533	0.000	0.007	**1.000**	0.024	0.000	4.924	0.000	5.321	0.000	0.282
l	**0.998**	1.566	0.000	0.009	0.975	0.023	0.000	0.622	0.000	5.057	0.000	0.272
m	0.997	1.542	0.000	0.007	**1.000**	0.025	0.000	3.271	0.000	5.081	0.000	0.278
n	0.949	1.544	0.000	0.007	**1.000**	0.022	0.000	2.420	**1.000**	6.039	0.000	0.274
o	0.999	1.535	0.000	0.007	**1.000**	0.026	0.000	0.596	0.000	3.417	0.000	0.275
Avg	0.980	1.52	0.467	0.007	0.989	0.024	0.580	2.725	0.514	5.683	0.562	0.269

Table 4.2: GMean and computation time (in seconds) for the 15 synthetic datasets, comparing ORBIT with five other anomaly detection methods.

The performance of **ORBIT** was exceptional across all synthetic datasets, achieving a GMean score of 0.945 or higher. Notably, ORBIT outperformed all other anomaly detection methods on the outlier local sparse uniform dataset (j) and outlier local sparse multimodal dataset (l). The average GMean score for the fifteen datasets was 0.98, comparable to the best-performing method, LOF, with a score of 0.989. The locations of the outlier instances detected by ORBIT in the synthetic datasets can be found in Appendix A. ORBIT exhibited a high level of confidence in identifying outlier instances, as indicated by the red highlights. However, there were instances identified as false positives, highlighted in green or orange. These false positives were primarily observed at the edges of the dimensional space, possibly due to the randomly selected reference points not being sufficiently representative in those areas. In terms of computation time, ORBIT performed reasonably fast compared to other well-developed Python libraries such as numpy, and sklearn for these synthetic datasets.

The **Median Standard Deviation** method showed satisfactory performance for datasets following uniform or Gaussian distributions, such as dataset (d), (e), (g), and (h). However, it failed to identify outliers in datasets with complex multimodal distributions like dataset (f) and (i). This limitation arises from the method's assumptions about the data distribution. Furthermore, the Median Standard Deviation method is incapable of detecting local outliers in datasets (j) to (o). Appendix B presents the positions of the identified outlier instances, highlighted in red.

The **LOF** method performed well across all synthetic datasets, achieving a GMean score of 0.989. It requires one parameter to be set, namely the number of nearest neighbors to compare. In this experiment, the number of nearest neighbors in LOF was set to 50 (2.5% of the total number of instances in the synthetic dataset). LOF exhibits high performance on synthetic datasets, primarily because the Local Reachability Density (LRD) of normal data points tends to be homogeneous. This homogeneity arises from the dense packing of normal data points in these datasets, which allows LOF to effectively distinguish outliers from normal instances with greater accuracy. However, it is worth noting that LOF was unable to detect a few instances in the local sparse outlier datasets (j) and (l). This occurred because these false negative instances had similar distances to their k-nearest neighbors, where the k-nearest normal instances also had these outlier instances as their neighbors. Appendix C illustrates the fifteen synthetic datasets, each displaying the positions of the identified outlier instances, highlighted in red.

The **K-means distance from centroid** method begins with k-means clustering, where the number of clusters is set to two. The centroid instance, located in the middle of the cluster, is then determined. As shown in Appendix D, the outlier instances detected by this method tend to be situated at the edges of the data instance clusters. This occurs because the centroid instance is calculated as the mean of all instances within the cluster, and instances located farther from the center are more likely to be identified as outliers. While this method can detect outliers in many datasets, it may be less effective in identifying local outliers located in the middle of the dimensional space.

Regarding the **K-means cluster size** method, it starts with k-means clustering using a large number of clusters, in this case, ten. The next step involves identifying the outlier instances from clusters containing fewer instances than a specific threshold. This technique performs well when the k-means clustering algorithm can group the outlier instances into a separate cluster, as observed in the dense outlier datasets (g), (h), and (n). However, for sparse outlier datasets, the k-means clustering algorithm tends to group the outlier instances into larger clusters, resulting in less accurate detection. Although this method can detect global outliers with uniform or Gaussian distributions, it struggles with complex multimodal distributions. Additionally, it does not perform well in identifying local outliers, making it less suitable for certain datasets. Appendix E displays the fifteen synthetic datasets, each showcasing the positions of the identified outlier instances, highlighted in red.

The **Isolation Forest** method successfully detected outliers in a majority of the global outlier datasets. However, it proved unsuccessful in identifying local outliers. As depicted in Appendix F, most of the false positive instances were located at the edges of the data. This occurrence arises from the isolation forest approach, where these instances at the edges are easier to isolate.

4.3 Experiment: Parameter Analysis

In this section, the examination of ORBIT aims to explore its robustness and assess the influence of different parameter settings. The experiment comprises two separate efficiency analyses. The first analysis concentrates on evaluating the robustness of ORBIT, specifically examining the impact of its randomness approach. The second analysis involves comparing the effectiveness of employing a high number of reference points with a low number of groups against using a low number of reference points with a high number of groups.

4.3.1 Robustness Analysis

To assess the impact of randomness on ORBIT, the algorithm was executed ten times, and the mean and standard deviation of the GMean, sensitivity, and specificity scores were measured. The result tables presented in this section indicate that the standard deviation of ORBIT is consistently low, measuring less than 0.01. This suggests that despite its random approach, ORBIT consistently produces similar GMean scores in each run.

4.3.2 Number of Reference Point and Group Analysis

The purpose of this experiment is to assess the robustness of ORBIT by investigating the impact of the number of reference points and groups. Two different experiment settings, RPG1 and RPG2, are employed. RPG1 corresponds to a configuration with a high number of reference points per group and a low number of groups, while RPG2 involves a low number of reference points per group and a high number of groups. The total number of reference points in the experiment is fixed at 25% of the total number of instances, which amounts to 500 for the synthetic datasets. Although both settings share a similar total number of reference points, each setting has its own limitations.

In the case of RPG1, having more reference points per group and fewer groups increases the likelihood of a group containing at least one outlier. Consequently, there may be a higher number of groups that contain outliers. Moreover, the higher number of reference points per group in RPG1 expands the group's turf reachability, covering a larger space, which can result in a higher rate of false negatives. RPG1 may also face challenges in identifying persistent outliers with sufficient confidence due to the limited number of groups providing independent empirical evaluation. On the other hand, RPG2 involves fewer reference points within each group, leading to reduced turf reachability and a higher rate of false positives for local outliers. To address this issue, the parameter for the standard deviation threshold for persistent outlier detection (o) can be increased to a higher value.

The results of the experiments conducted on two different parameter settings, RPG1 and RPG2, for the 15 synthetic datasets are presented in Table 4.3. Both parameter settings were found to be suitable for detecting outliers across all outlier types. RPG2 demonstrated superior performance compared to RPG1, with an average GMean score of 0.991 for RPG2 and 0.971 for RPG1 across the 15 synthetic datasets. Both RPG1 and RPG2 exhibited a high sensitivity score of 100% for a majority of the datasets, indicating their ability to correctly identify outlier instances. RPG1 showed a faster run time as it involved a lower number of groups to assess whether each instance was within or outside the reference turf. However, RPG1 exhibited a lower specificity score for all datasets, indicating a higher rate of false positives. To enhance the specificity performance, one possible step is to increase the standard deviation settings for persist outlier detection, such as setting it to a higher value (for example 3). This adjustment resulted in an improved GMean score of 0.996 ± 0.003 for RPG1.

4.4 Experiment: Data Characteristics Analysis

4.4.1 Heterogeneity in Data

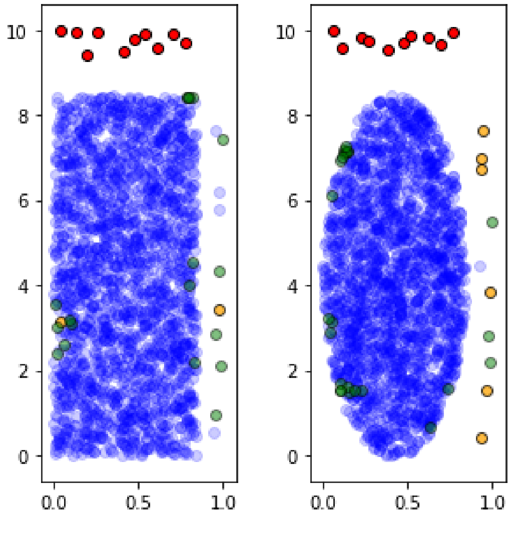
The objective of this experiment is to investigate the influence of different feature value scales on the performance of ORBIT. The synthetic dataset utilized in this experiment consists of two features with contrasting value scales. The first feature ranges from 0 to 1, while the second feature ranges from 0 to 10, which is ten times larger than the first feature. Table 4.4 presents the outcomes of the experiment for both normalized and non-normalized datasets. The results demonstrate that normalizing the data in ORBIT yields superior performance. The average GMean score for the normalized data across the 15 datasets was 0.988, whereas it was only 0.736 for the non-normalized data. The non-normalized dataset reveals that ORBIT successfully detected the outlier in the feature with the larger scale dimension (y-axis) for the outlier global sparse dataset. However, it failed to identify outliers in the feature with the smaller scale dimension (x-axis), as illustrated in Figure 4.2. This discrepancy arises due to the distance computation for the larger scale feature, which amplifies the distance between instances in the larger-scale feature and leads to a larger standard deviation. While outliers on the smaller-scale dimension may have larger values in that dimension, this alone is not enough to create a sufficiently large distance to exceed two standard deviations, as their values in the larger-scale dimension remain typical.

No	Dataset	ORBIT RPG1 (r=50, g=10, k=50, t=2, o=2)				ORBIT RPG2 (r=10, g=50, k=50, t=2, o=2)			
		GMean	Sensitivity	Specificity	Time (secs)	GMean	Sensitivity	Specificity	Time (secs)
a	No outlier uniform	0.959 ± 0.01	1.000 ± 0.00	0.920 ± 0.01	0.62	**0.981** ± 0.01	1.000 ± 0.00	0.963 ± 0.02	3.08
b	No outlier gaussian	0.962 ± 0.01	1.000 ± 0.00	0.925 ± 0.01	0.61	**0.979** ± 0.01	1.000 ± 0.00	0.958 ± 0.01	3.08
c	No outlier multimodal	0.957 ± 0.01	1.000 ± 0.00	0.916 ± 0.02	0.61	**0.983** ± 0.00	1.000 ± 0.00	0.967 ± 0.01	2.99
d	Outlier global sparse uniform	0.977 ± 0.01	1.000 ± 0.00	0.954 ± 0.01	0.58	**0.990** ± 0.00	1.000 ± 0.00	0.979 ± 0.01	3.16
e	Outlier global sparse gaussian	0.976 ± 0.01	1.000 ± 0.00	0.952 ± 0.02	0.58	**0.992** ± 0.00	1.000 ± 0.00	0.985 ± 0.01	3.19
f	Outlier global sparse multimodal	0.967 ± 0.01	1.000 ± 0.00	0.935 ± 0.01	0.61	**0.993** ± 0.00	1.000 ± 0.00	0.986 ± 0.01	3.18
g	Outlier global dense uniform	0.970 ± 0.01	1.000 ± 0.00	0.941 ± 0.02	0.61	**0.992** ± 0.00	1.000 ± 0.00	0.983 ± 0.01	3.21
h	Outlier global dense gaussian	0.974 ± 0.01	1.000 ± 0.00	0.948 ± 0.01	0.58	**0.994** ± 0.00	1.000 ± 0.00	0.988 ± 0.01	3.20
i	Outlier global dense multimodal	0.992 ± 0.00	1.000 ± 0.00	0.984 ± 0.01	0.64	**0.998** ± 0.00	1.000 ± 0.00	0.996 ± 0.00	3.22
j	Outlier local sparse uniform	0.972 ± 0.02	0.995 ± 0.02	0.949 ± 0.03	0.61	**0.994** ± 0.00	1.000 ± 0.00	0.988 ± 0.01	3.13
k	Outlier local sparse gaussian	0.975 ± 0.01	1.000 ± 0.00	0.950 ± 0.02	0.60	**0.994** ± 0.00	1.000 ± 0.00	0.987 ± 0.01	3.21
l	Outlier local sparse multimodal	0.971 ± 0.01	1.000 ± 0.00	0.942 ± 0.03	0.61	**0.992** ± 0.01	0.995 ± 0.02	0.990 ± 0.00	3.10
m	Outlier local dense uniform	0.971 ± 0.01	1.000 ± 0.00	0.943 ± 0.02	0.60	**0.992** ± 0.00	1.000 ± 0.00	0.983 ± 0.01	3.12
n	Outlier local dense gaussian	0.973 ± 0.01	1.000 ± 0.00	0.947 ± 0.03	0.60	**0.993** ± 0.00	1.000 ± 0.00	0.986 ± 0.01	3.15
o	Outlier local dense multimodal	0.978 ± 0.01	1.000 ± 0.00	0.957 ± 0.02	0.62	**0.996** ± 0.00	1.000 ± 0.00	0.993 ± 0.00	3.16
Average		0.971 ± 0.01	1.000 ± 0.00	0.944 ± 0.02	0.61	**0.991** ± 0.00	1.000 ± 0.00	0.982 ± 0.01	3.15

Table 4.3: Comparison of parameter settings between RPG1 (high number of reference points per group and low number of groups) and RPG2 (low number of reference points per group and high number of groups) on the 15 synthetic datasets.

No	Dataset Name	ORBIT with normalization			ORBIT without normalization		
		GMean	Sensitivity	Specificity	GMean	Sensitivity	Specificity
a	No outlier uniform	**0.976** ± 0.01	1.000 ± 0.00	0.953 ± 0.01	0.963 ± 0.01	1.000 ± 0.00	0.928 ± 0.02
b	No outlier gaussian	**0.976** ± 0.01	1.000 ± 0.00	0.952 ± 0.01	0.967 ± 0.01	1.000 ± 0.00	0.935 ± 0.02
c	No outlier multimodal	**0.976** ± 0.01	1.000 ± 0.00	0.953 ± 0.01	0.970 ± 0.01	1.000 ± 0.00	0.941 ± 0.01
d	Outlier global sparse uniform	**0.989** ± 0.01	1.000 ± 0.00	0.977 ± 0.01	0.942 ± 0.03	0.930 ± 0.05	0.955 ± 0.02
e	Outlier global sparse gaussian	**0.992** ± 0.00	1.000 ± 0.00	0.983 ± 0.01	0.970 ± 0.02	0.990 ± 0.02	0.950 ± 0.02
f	Outlier global sparse multimodal	**0.989** ± 0.01	1.000 ± 0.00	0.978 ± 0.01	0.973 ± 0.01	1.000 ± 0.00	0.946 ± 0.01
g	Outlier global dense uniform	**0.989** ± 0.01	1.000 ± 0.00	0.979 ± 0.01	0.984 ± 0.01	1.000 ± 0.00	0.969 ± 0.01
h	Outlier global dense gaussian	**0.992** ± 0.00	1.000 ± 0.00	0.985 ± 0.01	0.974 ± 0.01	1.000 ± 0.00	0.949 ± 0.02
i	Outlier global dense multimodal	**0.994** ± 0.00	1.000 ± 0.00	0.988 ± 0.01	0.972 ± 0.01	1.000 ± 0.00	0.944 ± 0.02
j	Outlier local sparse uniform	**0.991** ± 0.00	1.000 ± 0.00	0.982 ± 0.01	0.387 ± 0.21	0.205 ± 0.17	0.948 ± 0.02
k	Outlier local sparse gaussian	**0.989** ± 0.01	1.000 ± 0.00	0.979 ± 0.01	0.830 ± 0.09	0.735 ± 0.15	0.951 ± 0.02
l	Outlier local sparse multimodal	**0.990** ± 0.01	1.000 ± 0.00	0.979 ± 0.01	0.000 ± 0.00	0.000 ± 0.00	0.943 ± 0.01
m	Outlier local dense uniform	**0.991** ± 0.01	1.000 ± 0.00	0.982 ± 0.01	0.213 ± 0.28	0.130 ± 0.20	0.956 ± 0.02
n	Outlier local dense gaussian	**0.991** ± 0.01	1.000 ± 0.00	0.982 ± 0.01	0.811 ± 0.17	0.720 ± 0.24	0.952 ± 0.02
o	Outlier local dense multimodal	**0.994** ± 0.00	1.000 ± 0.00	0.988 ± 0.00	0.089 ± 0.14	0.030 ± 0.05	0.932 ± 0.02
Average		0.988 ± 0.00	1.000 ± 0.00	0.976 ± 0.01	0.736 ± 0.07	0.716 ± 0.06	0.947 ± 0.02

Table 4.4: Comparison of ORBIT's GMean, sensitivity, and specificity on synthetic datasets with and without normalization.



(a) (b)

Figure 4.2: ORBIT applied to the non-normalized synthetic dataset for detecting global sparse outliers. (a) Uniform distribution. (b) Gaussian distribution. Red circles represent outliers that would be reported by ORBIT as 4 standard deviations (sd); orange circles represent outliers at 2 sd; green circles represent outliers at 2 sd.

4.4.2 High dimensional data

Experimenting with high-dimensional data is essential for evaluating the performance and efficacy of algorithms in realistic and complex scenarios. High-dimensional data presents distinct challenges, including the curse of dimensionality, sparsity, and increased computation complexity. Assessing algorithms on high-dimensional datasets provides valuable insights into their scalability, robustness, and generalization abilities in contexts that closely resemble real-world applications.

In this experiment, additional irrelevant features were introduced to two synthetic datasets labeled as (h) and (n). The objective was to evaluate the effect of including different numbers of irrelevant features, specifically 2, 4, 16, and 32 features, on the algorithm's performance. The experiment maintained consistent parameter settings, including a fixed number of reference points (r) at 25, the number of groups (g) set to 20, the number of nearest distances (g) at 50, the standard deviation threshold for the turf reachability boundary (t) set to 2, and the standard deviation threshold for persistent outlier detection (o) set to 2. By systematically varying the number of irrelevant features while keeping other factors constant, the aim was to examine the algorithm's resilience and its ability to discern relevant information from noise in scenarios with increasing feature dimensions.

The results indicate that as the number of uniform random irrelevant features increased, the GMean score for outlier detection in the ORBIT algorithm decreased as shown in Table 4.5. Upon closer examination, it was observed that the sensitivity decreased while the specificity remained high, indicating that the algorithm struggled to identify outlier instances. The increase in feature dimensionality solely affected the pairwise distance measure. Another noteworthy point is that since the irrelevant data was generated completely randomly, following a theorem of Beyer et al. (1999), the higher the feature dimensionality, the more challenging it becomes for the algorithm to detect outliers. Specifically, it was shown by Beyer et al. (1999) that, as dimensionality increases, the distance to the nearest data point approaches the distance to the farthest data point. Empirical results on real and synthetic datasets demonstrate that this effect occurs even with as few as 10 dimensions. Nevertheless, real-life scenarios often involve specific distribution of features. Therefore, experiments that introduce completely random and uniform irrelevant data may yield outcomes that are potentially exaggerated compared to real-life datasets.

4.4.3 Large number of instances

The purpose of this experiment is to assess the time complexity performance by increasing the number of instances. The experiment uses 2,000, 10,000, 100,000, 500,000, and 1 million instances, which are based on the outlier global dense Gaussian dataset containing 1% outlier instances. The parameter settings remain consistent, with the number of reference points (r) set to 25, the number of groups (g) set to 20, the number of nearest neighbours (k) set to 50, the standard deviation threshold for the turf reachability boundary (t) set to 2, and the standard deviation threshold for persist outlier detection (o)

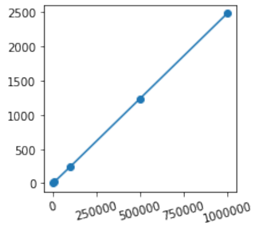
No	Dataset Name	Number of features	ORBIT (r=25, g=20, k=50, t=2, o=2)			
			GMean	Sensitivity	Specificity	Time (secs)
h	Outlier	2+0	**0.992** ± 0.00	1.000 ± 0.00	0.985 ± 0.01	1.53
	global	2+2	0.984 ± 0.00	1.000 ± 0.00	0.969 ± 0.01	1.34
	dense	2+4	0.982 ± 0.01	1.000 ± 0.00	0.964 ± 0.01	1.16
	gaussian	2+8	0.937 ± 0.02	0.945 ± 0.04	0.930 ± 0.01	0.89
		2+16	0.659 ± 0.06	0.460 ± 0.09	0.952 ± 0.01	0.63
		2+32	0.325 ± 0.06	0.110 ± 0.04	0.996 ± 0.00	0.46
n	Outlier	2+0	**0.991** ± 0.01	1.000 ± 0.00	0.982 ± 0.01	1.54
	local	2+2	0.781 ± 0.11	0.645 ± 0.17	0.966 ± 0.01	1.40
	dense	2+4	0.021 ± 0.06	0.005 ± 0.02	0.902 ± 0.01	1.212
	gaussian	2+8	0.022 ± 0.07	0.005 ± 0.02	0.946 ± 0.01	1.21
		2+16	0.000 ± 0.00	0.000 ± 0.00	0.991 ± 0.00	0.589
		2+32	0.000 ± 0.00	0.000 ± 0.00	1.000 ± 0.00	0.471

Table 4.5: ORBIT performance metrics (GMean, sensitivity, specificity) and computation time for high-dimensional data on global and local dense Gaussian synthetic datasets. The evaluation is conducted on two relevant features and varying numbers of irrelevant features (2, 4, 8, 16, 32).

set to 2.

The computation time results, as shown in Table 4.6, indicate that the computation time for both the outlier global (h) and outlier local (n) dense Gaussian datasets increases proportionally with the number of instances. This can be attributed to the fact that the number of pairwise comparisons required depends on the number of reference points to each instance. The number of pairwise distance measures needed for the five different instance sizes is 1 million, 5 million, 50 million, 250 million, and 500 million, respectively.

No	Dataset Name	Instance count	ORBIT
			Time (secs)
h	Outlier global dense gaussian	2,000 10,000 100,000 500,000 1,000,000	1.53 19.63 239.29 1,235.23 2,481.25
n	Outlier local dense gaussian	2,000 10,000 100,000 500,000 1,000,000	1.54 19.64 241.79 1,221.21 2,468.76



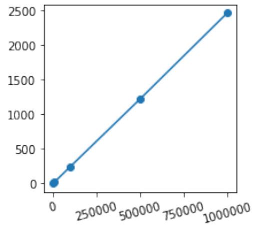


Table 4.6: ORBIT computation time (in seconds) for different instance sizes on global (h) and local (n) dense Gaussian outlier datasets.

4.5 Experiment: Benchmark Data

This section presents the results and discussion of ORBIT and ten other anomaly detection methods described earlier. Table 4.7 provides a summary of the GMean scores for the 19 benchmark datasets. The computation time (in seconds) for the 19 benchmark datasets can be found in Appendix G for reference. Among the outlier detection techniques evaluated, ORBIT outperforms the others in a greater number of benchmark datasets, specifically in 6 out of 19 datasets. The average GMean score across the benchmark datasets demonstrates that ORBIT achieves the highest score of 0.705, followed by the ROD method with a score of 0.688 and with 2 datasets achieving the highest GMean score. While the ABOD method obtains the highest GMean score on three datasets. When considering the top-two and three rankings, ORBIT stands out ~~with seven additional datasets~~ where it achieves the top-two or three GMean scores on these additional datasets: Glass, WBC, Vowels, Letter, Smtp, Musk, and Satimage-2.

ECOD and HBOS each achieve the top-two or three GMean scores on six datasets, while Median SD achieves the top-two or three GMean score on five datasets.

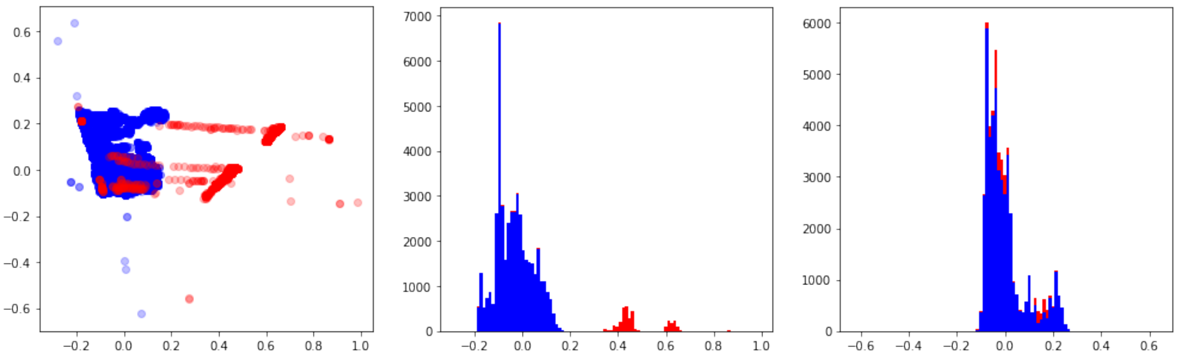
Dataset	ORBIT	Median SD	LOF	Cluster distance to centroid	Cluster instance size	IF	ABOD	COPOD	ECOD	HBOS	ROD
Wine	**0.852**	0.748	-	0.31	-	-	-	0.751	0.524	0.608	0.683
Lympho	**0.986**	0.252	0.407	0.91	0.577	0.577	0.763	0.968	0.968	0.968	0.88
Glass	0.612	0.6	**0.679**	0.451	0.548	-	0.447	0.316	0.448	0.316	0.448
WBC	0.822	0.818	0.798	0.301	0.82	0.377	0.715	0.817	0.759	**0.846**	0.817
Vowels	0.841	0.607	0.424	0.462	-	0.345	**0.903**	0.232	0.466	0.485	0.425
Letter	0.568	0.352	0.141	0.392	-	-	**0.73**	0.315	0.315	0.355	0.300
Musk	0.997	0.094	-	**1.000**	**1.000**	0.565	0.133	0.827	0.856	0.964	
Thyroid	0.899	0.83	0.888	0.705	0.684	0.463	-	0.854	**0.939**	0.912	**0.939**
Speech	0.219	0.064	-	0.128	-	-	**0.719**	0.298	0.321	0.298	
Optdigits	**0.628**	0.373	-	0.196	-	0.115	0.378	0.268	0.257	**0.628**	0.497
Satimage-2	0.964	0.888	0.967	**0.977**	0.927	0.896	0.605	0.920	0.906	0.927	0.869
Pendigits	**0.959**	0.878	0.209	0.882	-	0.422	0.419	0.704	0.776	0.821	0.825
Annthyroid	0.522	0.512	0.517	0.373	0.262	0.296	-	0.510	0.572	0.547	**0.613**
Mnist	**0.556**	0.224	0.500	0.370	-	0.241	0.555	0.488	0.423	0.389	
Mammography	0.560	0.753	0.430	0.270	0.702	0.401	-	**0.833**	0.831	0.711	0.348
Shuttle	0.268	0.925	0.390	0.144	0.920	0.373	0.452	**0.976**	0.974	0.967	0.954
Smtp	0.824	**0.833**	0.820	0.799	0.178	-	-	0.794	0.794	0.799	0.775
ForestCover	0.312	**0.785**	0.110	0.619	-	0.286	0.589	0.683	0.785	0.556	0.691
Http	**0.997**	0.98	0.230	0.995	0.996	0.996	-	0.951	0.951	0.963	0.950
Average GMean	**0.705**	0.606	0.395	0.541	0.413	0.334	0.390	0.658	0.677	0.687	0.688
Highest GMean (count)	**6**	2	1	2	1	0	3	2	2	2	2
2nd Highest GMean (count)	5	0	1	1	1	0	1	2	6	1	0
3rd Highest GMean (count)	2	5	2	0	1	0	0	1	0	5	2

Table 4.7: Summary Results of the Benchmark Datasets. The highest GMean for each dataset is highlighted in bold.

Upon a detailed investigation, it was observed that ORBIT performs poorly on the shuttle dataset, achieving a GMean of 0.268. In contrast, the COPOD method achieves the highest GMean of 0.976, while Median Standard Deviation method achieves of 0.925 on the same dataset. The discrepancy in performance might be attributed to the fact that as the number of features increases, the distance to the nearest data point approaches the distance to the farthest data point, as stated by Beyer et al. (1999). Whereas, the Median Standard Deviation method considers each feature independently, using a different notion of distance than that described by Beyer et al. (1999), which allows it to circumvent the limitations of their theorem. However, the shuttle dataset has only nine features, which is likely too few for the effect described by the theorem of Beyer et al. (1999) to be observable.

To gain further insights, the nine features of the shuttle dataset were transformed into two components using Principal Component Analysis (PCA) (Tipping and Bishop, 1999). Figure 4.3 illustrates the resulting PCA components, displaying the data distribution with normal instances in blue and outlier instances in red. The outliers comprise 7% of the total instances, amounting to 3,511 observations.

Upon examining the PCA plots, it becomes evident that the outliers are distinguishable primarily based on the first component, PC1. However, ORBIT encounters challenges when attempting to identify these clustered outlier instances using a low number of nearest neighbors used to compute the reference point turf reachability boundary. This difficulty arises due to the concentration of closely clustered outliers. The default parameter value in ORBIT sets the number of nearest neighbors to 2.5% (or 2,455 instances), which may not be sufficient to accurately detect these outliers. Consequently, ORBIT ends up selecting some outlier instances as reference points, leading to other outlier instances falling within the turfs of these outlier reference points.



(a) (b) (c)

Figure 4.3: PCA results for the shuttle benchmark dataset. (a) Scatter plot of the two components. (b) Distribution of PC1. (c) Distribution of PC2. Blue color represents normal instances, while red color represents outlier instances.

As a result, two potential recommendations arise. The first suggestion is to train ORBIT exclusively on data that exhibits a high level of confidence in being normal instances. Alternatively, if the percentage of outliers in the dataset is known, the number of nearest neighbors can be set to at least twice the percentage of outliers. Taking the latter approach, ORBIT was executed with a k nearest neighbor parameter of 20% (9,819 instances), which corresponds to approximately three times the total number of outlier instances. This adjustment significantly improves ORBIT's performance, yielding a GMean of 0.970, along with sensitivity and specificity scores of 0.941 and 0.999, respectively. This result is comparable to the performance of COPOD, which achieved a GMean of 0.976. Furthermore, it outperforms the Median Standard Deviation approach, which attained a GMean of 0.925, sensitivity of 0.982, and specificity of 0.871.

Examining the results from Table 4.7, the Median Standard Deviation method exhibits strong performance in larger instance datasets (consisting of more than 10,000 instances), outperforming on two out of the five datasets compared to other methods. On the largest dataset (HTTP), the Median Standard Deviation method achieves a high GMean score of 0.980, which is comparable to the highest GMean score of 0.997 achieved by ORBIT.

It is important to highlight that while LOF demonstrated superior performance on synthetic datasets as shown in Table 4.2, its performance was much weaker when evaluated on benchmark datasets. This observation suggests that the synthetic datasets might not adequately represent the diverse range of data distributions found in real-world scenarios. Specifically, it is possible that the normal instances in the synthetic datasets are excessively densely packed, deviating from the distribution patterns typically encountered in real data.

4.6 Comparison: ORBIT and Other Approaches

The comparison between ORBIT and LOF reveals an advantage in ORBIT's utilization of nearest neighbors compared to LOF. While LOF considers a fixed set of k-nearest neighbors specific to each data instance, ORBIT employs a fixed set of k-nearest neighbors exclusively for reference points, which are more likely to be normal instances. ORBIT's evaluation does not rely on the nearest neighbors of the data instance being analyzed. Instead, it focuses on determining whether the data instance falls within the turf of multiple reference point groups. Empirically, as demonstrated in Table 4.7, this approach demonstrates greater robustness. However, further investigation is required to establish the exact theoretical rationale behind these results.

Another difference is that the LOF assumes outliers as regions with lower density compared to their neighbors, utilizing this density differential for outlier reporting. However, in datasets with uneven or varying density distributions, LOF may encounter challenges in accurately identifying outliers. In such cases, outliers may not necessarily exhibit lower density but could still display anomalous behavior relative to their local surroundings. On the other hand, ORBIT relies less on density and instead focuses on the disparity in the likelihood of a normal instance falling within the turf of reference points (which are more likely to be normal instances) versus the likelihood of an outlier instance being within the turf of reference points. By leveraging this difference in probabilities, ORBIT offers a different perspective on outlier detection that may be advantageous in scenarios where density-based approaches struggle to capture anomalies accurately.

When comparing ORBIT with clustering-based outlier detection methods, ORBIT demonstrates a clear advantage in terms of robustness and outlier detection performance. Unlike the K-means clustering method, where the initial seed selection plays a critical role and can significantly impact the final results, ORBIT's outlier detection process is designed to be more stable and reliable. ORBIT achieves this by utilizing multiple independent groups, each responsible for detecting local outliers within its own group. The true outliers are then determined based on the number of times an instance is labeled as a local outlier across different groups. This approach ensures a more consistent and accurate identification of outliers in the dataset.

5 Summary and Future Work

5.1 Summary

This research has focused on comprehensively understanding various anomaly detection techniques and their effectiveness in accurately identifying anomalies. By considering domain-related, data-related, model-related, and result-related factors, valuable insights have been gained for the implementation of reliable and effective anomaly detection solutions. The research has emphasized domain-specific characteristics, the evaluation of data attributes, critical assessment of techniques and parameters, and the selection of appropriate evaluation metrics to optimize anomaly detection performance.

As a research contribution, a novel outlier detection technique called Outlier Reporting By Inference on Turfs (ORBIT) has been developed. ORBIT exhibits several strengths that enhance its effectiveness in detecting outliers. By leveraging the assumption that datasets primarily consist of normal instances with few outliers, ORBIT's reference selection component enables the algorithm to utilize representative characteristics from a small sample of references. Additionally, ORBIT uses the turf inference component to define the reachability boundary and determine typical distances of normal instances from reference points. The grouping component ensures comprehensive coverage of normal regions by dividing references into dispersed groups. In scenarios where a normal instance falls outside multiple groups, the persistent outliers component identifies instances consistently outside turfs of most groups and treats them as outliers. By analyzing the distribution of instances falling outside the turfs of a group, ORBIT establishes a threshold based on the median plus n standard deviations. This integrated approach of ORBIT enhances computation efficiency, promotes better understanding, and achieves high accuracy in anomaly detection, making it a robust solution for anomaly detection.

Extensive experiments and evaluations were conducted to assess the performance and robustness of the ORBIT outlier detection solution. Various factors including parameter settings, irrelevant features, data normalization, high-dimensional data, and benchmark dataset performance were thoroughly evaluated. The results consistently demonstrate the exceptional performance of ORBIT, surpassing other anomaly detection methods in most cases. Out of the 19 evaluated datasets, ORBIT achieves the highest GMean score on 6 datasets and an average GMean score of 0.705. The closest competitor achieves the highest GMean score on 2 datasets and an average score of 0.688. ORBIT initially exhibited poor performance on the shuttle benchmark dataset. Upon further investigation, it was discovered that the dataset contains a significant number of outlier instances that are densely clustered, exceeding the assumption on the number of nearest neighbors. By adjusting the parameter settings of ORBIT and increasing the number of nearest neighbors to three times the count of outliers, a significant improvement was observed. The algorithm's GMean score notably increased to 0.970, comparable with the score of 0.976 obtained by the next best approach. This successful parameter optimization demonstrates ORBIT's adaptability and its ability to outperform other methods in outlier detection tasks.

Overall, this research provides valuable insights for implementing reliable and effective anomaly detection solutions. The findings highlight ORBIT's effectiveness as a reliable outlier detection technique, particularly in challenging scenarios.

5.2 Future Work

While this research has provided valuable insights into anomaly detection techniques and the effectiveness of the ORBIT solution, there are several avenues for future exploration and improvement. The following areas can be considered for future work:

Algorithmic Enhancements: Refinement and optimization of the ORBIT algorithm can be explored to enhance its performance as shown in Table 3.1 highlighted in blue colours. This includes performing clustering prior to reference point selection. Incorporating clustering prior to reference point selection in the ORBIT algorithm allows for the identification of distinct clusters representing different normal regions in the data. By selecting representative reference points from these clusters, the algorithm focuses on regions with a higher concentration of normal instances, thereby improving the quality of the reference points.

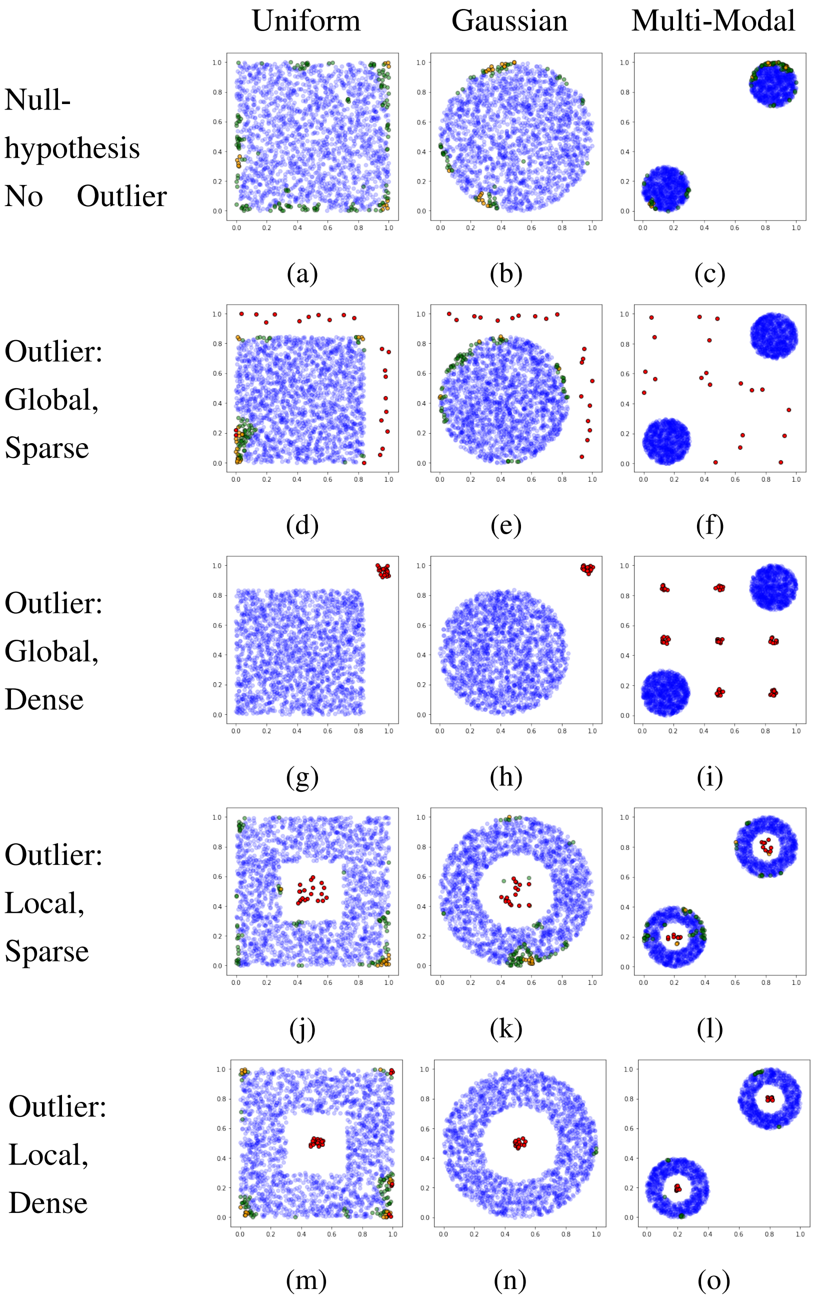
ORBIT Guidelines for Achieving Optimal Anomaly Detection Performance:

Future work should prioritize the development of comprehensive guidelines for effectively utilizing ORBIT, optimizing its parameter settings according to dataset characteristics, and conducting thorough result evaluations. These guidelines will serve as valuable resources to users, enabling them to adapt ORBIT to various dataset characteristics and topology, and make informed decisions during its application. A fundamental step is to gain a deep understanding of the dataset's characteristics. For instance, in the case of the shuttle benchmark dataset, which includes three dense clusters with two smaller ones, adjusting the number of nearest neighbors can significantly improve the results. By thoroughly evaluating dataset characteristics, including size, type, distribution, patterns, and quality issues, users can obtain critical insights that facilitate the selection of appropriate parameter settings for ORBIT. The adaptation of parameter settings is another crucial aspect that the guidelines should address. Recommendations should focus on adjusting parameters based on the specific dataset characteristics. For datasets with a high proportion of outliers, it might be necessary to increase the number of nearest neighbors. By providing guidance on adapting parameter settings to different dataset characteristics, the performance of ORBIT can be optimized for specific datasets.

The guidelines should also address the evaluation of results. They should outline suitable evaluation metrics and methodologies to assess ORBIT's performance, such as sensitivity, specificity, and GMean. Additionally, the guidelines should provide insights into interpreting the results, helping users understand the implications of high specificity or high sensitivity in the context of anomaly detection. By providing comprehensive guidelines encompassing dataset evaluation, parameter adaptation, and result evaluation, users can effectively utilize ORBIT and optimize its parameter settings for reliable and accurate anomaly detection. These guidelines will serve as a valuable resource for practitioners, empowering them to leverage the full potential of ORBIT for their specific anomaly detection tasks.

Another future work, **the real-world implementation and deployment** of ORBIT can greatly benefit from close collaboration with industry partners. By engaging in such collaborations, researchers can leverage the domain-specific expertise and insights of industry professionals, incorporating industry knowledge into the algorithm's development and ensuring its relevance and effectiveness in specific application areas. Collaborating with industry stakeholders also allows for tailoring the algorithm to different industries, addressing industry-specific anomalies, and exploring new applications beyond the initial scope. This collaborative approach, combining academic research with industry expertise, will drive the algorithm's applicability, address industry-specific challenges, and create solutions with a meaningful impact in real-world settings. Strong partnerships with industry stakeholders will provide valuable insights, data, and resources necessary to validate the algorithm's performance in real-world scenarios and integrate it into practical anomaly detection systems, facilitating its successful adoption and utilization in diverse industry contexts.

A Appendix A



(a) (b) (c)

(d) (e) (f)

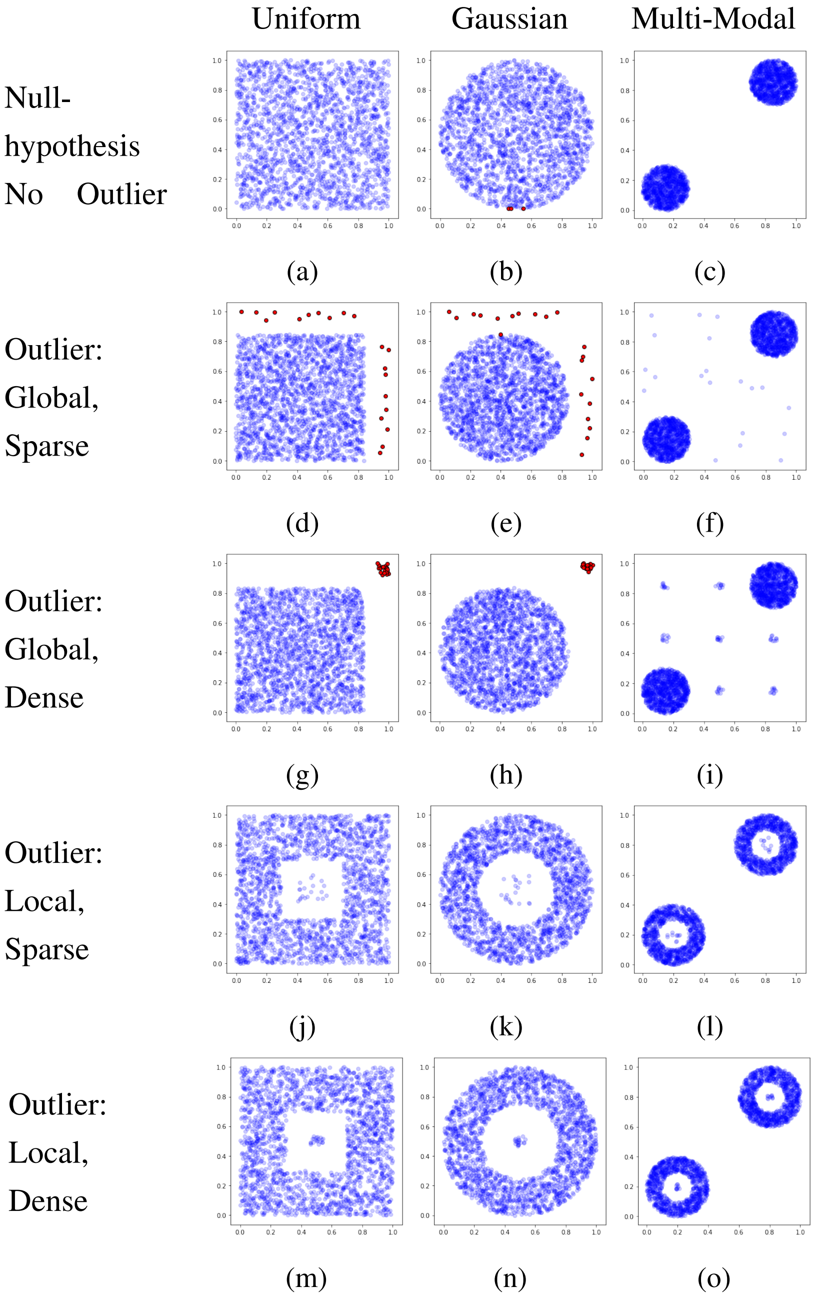
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure A.1: **ORBIT - Synthethic Datasets**. The location of the outlier instances reported by ORBIT with different confidence level highlighted in green (+2 sd), orange (+3 sd) and red (+4 sd) colour coding

B Appendix B



(a) (b) (c)

(d) (e) (f)

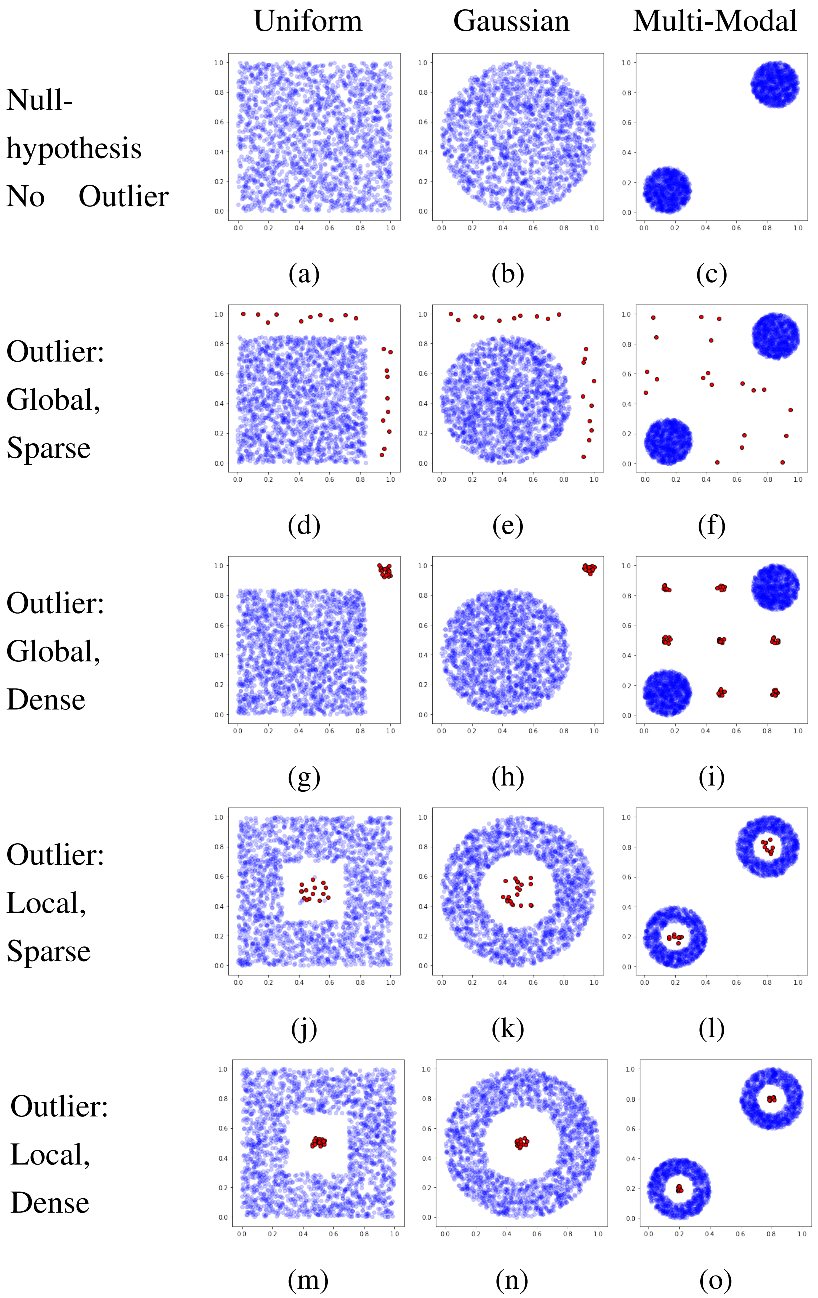
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure B.1: **Median Standard Deviation - Synthethic Dataset**. The result of Median Standard Deviation method in detecting outlier for fifteen synthethic dataset. The result shows the location of the reported normal instances (highlighted in blue) and reported outlier instances (highlighted in red)

C Appendix C



(a) (b) (c)

(d) (e) (f)

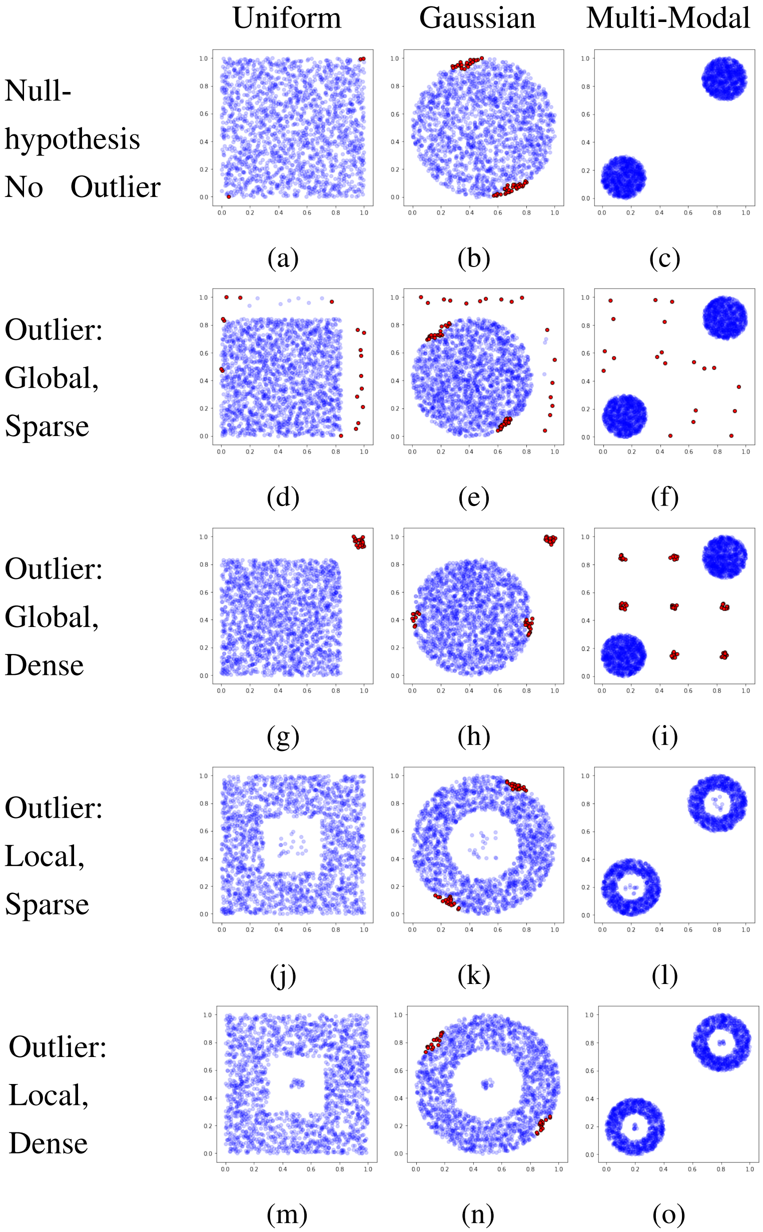
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure C.1: **Local Outlier Factor (LOF) - Synthethic Dataset**. The result of LOF method in detecting outlier for fifteen synthethic dataset. The result shows the location of the reported normal instances (highlighted in blue) and reported outlier instances (highlighted in red)

D Appendix D



(a) (b) (c)

(d) (e) (f)

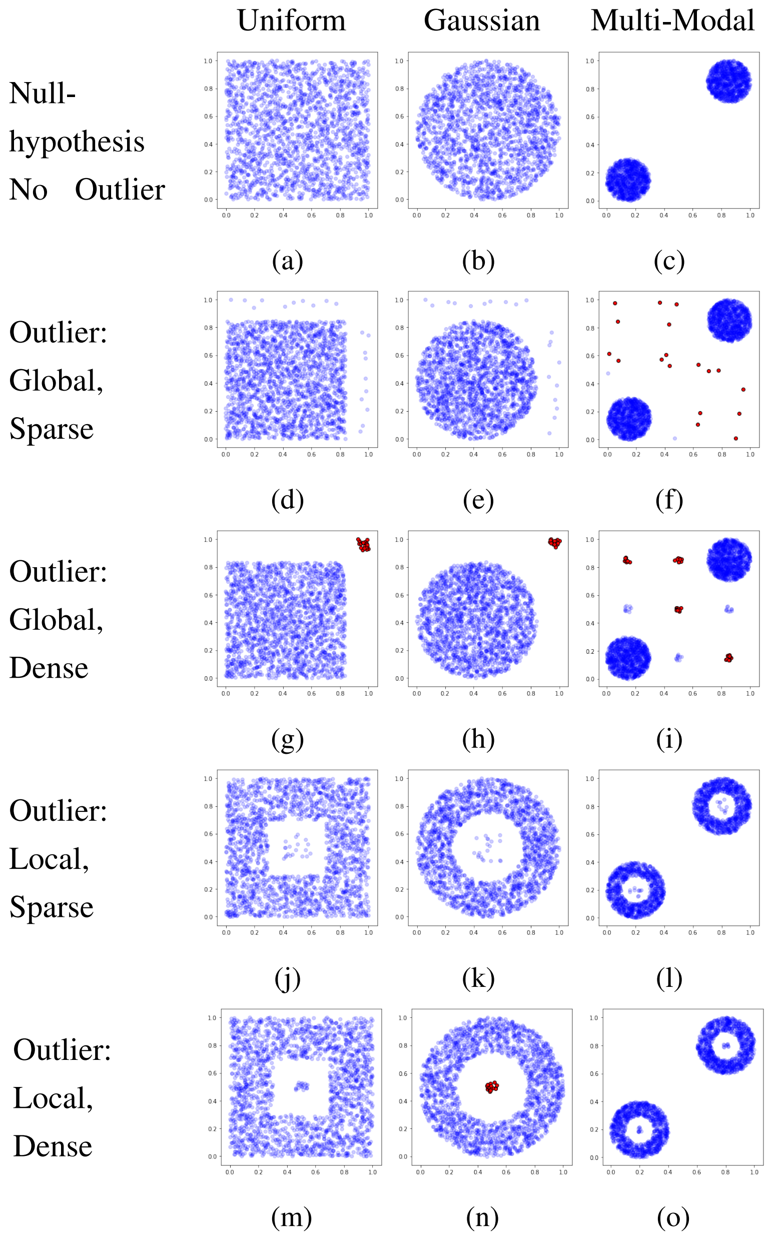
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure D.1: **K-means Clustering, outlier based on distance to centroid - Synthethic Dataset**. The result of K-means clustering method in detecting outlier for fifteen synthethic dataset, where the outlier is determine based on the distance to centroid. The result shows the location of the reported normal instances (highlighted in blue) and reported outlier instances (highlighted in red)

E Appendix E



(a) (b) (c)

(d) (e) (f)

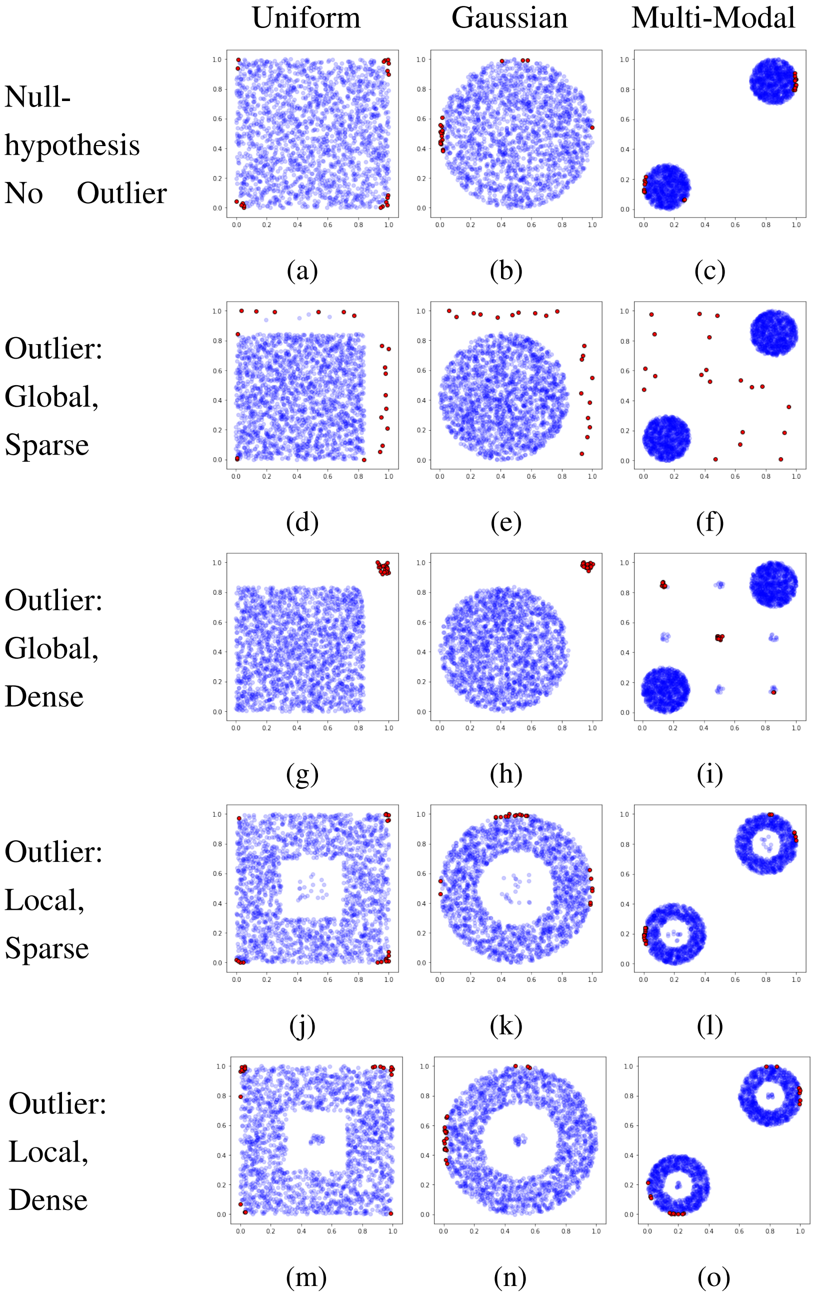
(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure E.1: **K-means Clustering, outlier based cluster size - Synthethic Dataset**. The result of K-means clustering method in detecting outlier for fifteen synthethic dataset, where the outlier is determine based on the size of the cluster. The result shows the location of the reported normal instances (highlighted in blue) and reported outlier instances (highlighted in red)

F Appendix F



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure F.1: **Isolation Forest (IF) - Synthethic Dataset**. The result of IF method in detecting outlier for fifteen synthethic dataset. The result shows the location of the reported normal instances (highlighted in blue) and reported outlier instances (highlighted in red)

G Appendix G

				Cluster	Cluster						
		Median		distance	instance						
Dataset	ORBIT	SD	LOF	to centroid	size	IF	ABOD	COPOD	ECOD	HBOS	ROD
Wine	0.04	0.01	0	0.02	0.05	0.18	0.04	0.05	0.01	0.05	2.45
Lympho	0.04	0.01	0	0.03	0.04	0.23	0.07	0.08	0.01	0.09	1.91
Glass	0.04	0.01	0	0.02	0.04	0.18	0.06	0.07	0.01	0.07	1.33
WBC	0.12	0.02	0.01	0.63	0.52	0.27	0.17	0.19	0.02	0.2	94.19
Vowels	0.74	0.04	0.38	1.22	7.42	0.23	0.42	0.44	0.02	0.44	16.26
Letter	1.18	0.08	0.06	1.16	6.88	0.27	0.59	0.64	0.04	0.65	119.22
Musk	3.59	0.82	0.25	1.11	3.16	0.47	2.69	3.27	0.58	3.33	
Thyroid	3.24	0.08	0.09	1.19	7.15	0.3	1.01	1.04	0.03	1.05	7.2
Speech	2.79	2.72	0.38	1.37	9.97	0.88	10.4	12.37	2	12.52	
Optdigits	4.07	0.79	0.61	1.31	10.49	0.49	4.03	4.25	0.22	4.28	3458.4
Satimage-2	6.62	0.37	0.62	1.28	5.11	0.43	2.71	2.89	0.18	2.91	1563.95
Pendigits	6.87	0.33	0.97	2.1	8.02	0.52	2.93	3.07	0.14	3.08	245.27
Annthyroid	6.03	0.24	0.2	2.87	10.34	0.45	2.91	2.97	0.06	4.37	10.36
Mnist	5.22	2.01	1.29	1.45	19.75	0.67	8.34	8.74	0.4	8.78	
Mammography	9.67	0.67	0.34	1.02	7.44	0.56	2.94	3.01	0.07	3.02	12.81
Shuttle	56.37	5.69	2.57	1.53	3.92	1.91	14.74	15.19	0.45	15.23	58.58
Smtp	465.12	6.19	2.4	1.61	5.06	3.28	23.39	23.8	0.41	23.87	4.19
ForestCover	786.98	422.92	30.89	6.36	20.19	9.55	88.55	92.63	4.07	92.88	1616.34
Http	4,954.37	167.06	216.59	5.73	6.48	18.23	524.69	527.2	2.52	527.62	13.78
Average GMean	332.27	32.11	13.56	1.68	6.95	2.06	36.35	36.94	0.59	37.08	451.64

Table G.1: Summary Results of Computation Time (in seconds) for 19 Benchmark Datasets.

References

ACM. 2023. “Computing Classification System.” (Accessed 01/03/2023). Available from: https://dl.acm.org/ccs.

Ahmed, Mohiuddin, Mahmood, Abdun Naser, and Islam, Md Rafiqul. 2016. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems* 55:278–288.

Almardeny, Yahya, Boujnah, Noureddine, and Cleary, Frances. 2020. A novel outlier detection method for multivariate data. *IEEE Transactions on Knowledge and Data Engineering* 34 (9): 4052–4062.

Angin, Pelin, Bhargava, Bharat, and Ranchal, Rohit. 2019. *Big data analytics for cyber security.*

Auger, Nicolas, Jugé, Vincent, Nicaud, Cyril, and Pivoteau, Carine. 2018. On the worst-case complexity of TimSort. *arXiv preprint arXiv:1805.08612.*

Beyer, Kevin, Goldstein, Jonathan, Ramakrishnan, Raghu, and Shaft, Uri. 1999. “When is “nearest neighbor” meaningful.” In *Database Theory—ICDT’99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7,* 217–235. Springer.

Boushey, Carol J, Beresford, Shirley AA, Omenn, Gilbert S, and Motulsky, Arno G. 1995. A quantitative assessment of plasma homocysteine as a risk factor for vascular disease: probable benefits of increasing folic acid intakes. *JAMA* 274 (13): 1049–1057.

Breunig, Markus M, Kriegel, Hans-Peter, Ng, Raymond T, and Sander, Jörg. 2000. “LOF: identifying density-based local outliers.” In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data,* 93–104.

Chabchoub, Yousra, Togbe, Maurras Ulbricht, Boly, Aliou, and Chiky, Raja. 2022. An in-depth study and improvement of Isolation Forest. *IEEE Access* 10:10219–10237.

Chalapathy, Raghavendra and Chawla, Sanjay. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407.*

Chandola, Varun, Banerjee, Arindam, and Kumar, Vipin. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41 (3): 1–58.

Chawla, Sanjay and Gionis, Aristides. 2013. “k-means–: A unified approach to clustering and outlier detection.” In *Proceedings of the 2013 SIAM International Conference on Data Mining,* 189–197. SIAM.

Dahmen, Jessamyn and Cook, Diane J. 2021. Indirectly supervised anomaly detection of clinically meaningful health events from smart home data. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12 (2): 1–18.

Dave, Dhwani and Varma, Tanvi. 2014. A review of various statistical methods for outlier detection. *International Journal of Computer Science & Engineering Technology (IJCSET)* 5 (2): 137–140.

Estiri, Hossein, Klann, Jeffrey G, and Murphy, Shawn N. 2019. A clustering approach for detecting implausible observation values in electronic health records data. *BMC Medical Informatics and Decision Making* 19:1–16.

Foorthuis, Ralph. 2021. On the nature and types of anomalies: A review of deviations in data. *International Journal of Data Science and Analytics* 12 (4): 297–331.

Friedman, Jerome H, Bentley, Jon Louis, and Finkel, Raphael Ari. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3 (3): 209–226.

Goldstein, Markus and Dengel, Andreas. 2012. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* 1:59–63.

Goldstein, Markus and Uchida, Seiichi. 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS ONE* 11 (4): e0152173.

Han, Songqiao, Hu, Xiyang, Huang, Hailiang, Jiang, Minqi, and Zhao, Yue. 2022. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems* 35:32142–32159.

Hariri, Sahand, Kind, Matias Carrasco, and Brunner, Robert J. 2019. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering* 33 (4): 1479–1489.

Kamalov, Firuz and Leung, Ho Hon. 2020. Outlier detection in high dimensional data. *Journal of Information & Knowledge Management* 19 (01): 2040013.

Kandanaarachchi, Sevvandi, Muñoz, Mario A, Hyndman, Rob J, and Smith-Miles, Kate. 2020. On normalization and algorithm selection for unsupervised outlier detection. *Data Mining and Knowledge Discovery* 34 (2): 309–354.

Kriegel, Hans-Peter, Schubert, Matthias, and Zimek, Arthur. 2008. "Angle-based outlier detection in high-dimensional data." In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 444–452.

Leys, Christophe, Ley, Christophe, Klein, Olivier, Bernard, Philippe, and Licata, Laurent. 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology* 49 (4): 764–766.

Li, Zheng, Zhao, Yue, Botta, Nicola, Ionescu, Cezar, and Hu, Xiyang. 2020. “COPOD: copula-based outlier detection.” In *2020 IEEE International Conference on Data Mining (ICDM),* 1118–1123. IEEE.

Li, Zheng, Zhao, Yue, Hu, Xiyang, Botta, Nicola, Ionescu, Cezar, and Chen, George. 2022. ECOD: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering.*

Liu, Fei Tony, Ting, Kai Ming, and Zhou, Zhi-Hua. 2008. “Isolation forest.” In *2008 Eighth IEEE International Conference on Data Mining,* 413–422. IEEE.

Lloyd, Stuart. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28 (2): 129–137.

Maurya, Chandresh Kumar, Toshniwal, Durga, and Venkoparao, Gopalan Vijendran. 2016. Online sparse class imbalance learning on big data. *Neurocomputing* 216:250–260.

McClish, Donna Katzman. 1989. Analyzing a portion of the ROC curve. *Medical decision making* 9 (3): 190–195.

Ngo, DuyHoa and Veeravalli, Bharadwaj. 2015. “Design of a real-time morphology-based anomaly detection method from ecg streams.” In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM),* 829–836. IEEE.

ODDS Outlier Detection Data Sets - odds.cs.stonybrook.edu. http://odds.cs.stonybrook.edu/. [Accessed 02-May-2023].

Poon, Lex, Farshidi, Siamak, Li, Na, and Zhao, Zhiming. 2021. “Unsupervised anomaly detection in data quality control.” In *2021 IEEE International Conference on Big Data (Big Data),* 2327–2336. IEEE.

Ramaswamy, Sridhar, Rastogi, Rajeev, and Shim, Kyuseok. 2000. “Efficient algorithms for mining outliers from large data sets.” In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data,* 427–438.

Riahi-Madvar, Mahboobeh, Azirani, Ahmad Akbari, Nasersharif, Babak, and Raahemi, Bijan. 2021. A new density-based subspace selection method using mutual information for high dimensional outlier detection. *Knowledge-Based Systems* 216:106733.

Rubin, Donald B. 1976. Inference and missing data. *Biometrika* 63 (3): 581–592.

Samara, Mustafa Al, Bennis, Ismail, Abouaissa, Abdelhafid, and Lorenz, Pascal. 2022. A survey of outlier detection techniques in IoT: review and classification. *Journal of Sensor and Actuator Networks* 11 (1): 4.

Seo, Jinwook and Shneiderman, Ben. 2004. “A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections.” In *IEEE Symposium on Information Visualization,* 65–72. IEEE.

Shao, Chen, Du, Xusheng, Yu, Jiong, and Chen, Jiaying. 2022. Cluster-based improved isolation forest. *Entropy* 24 (5): 611.

Siddiqui, Md Amran, Stokes, Jack W, Seifert, Christian, Argyle, Evan, McCann, Robert, Neil, Joshua, and Carroll, Justin. 2019. “Detecting cyber attacks using anomaly detection with explanations and expert feedback.” In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2872–2876. IEEE.

Sikder, Md Nazmul Kabir and Batarseh, Feras A. 2023. Outlier detection using AI: a survey. *AI Assurance,* 231–291.

Smiti, Abir. 2020. A critical overview of outlier detection methods. *Computer Science Review* 38:100306.

Thudumu, Srikanth, Branch, Philip, Jin, Jiong, and Singh, Jugdutt. 2020. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data* 7:1–30.

Tipping, Michael E and Bishop, Christopher M. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61 (3): 611–622.

Umar, Mubarak Albarka and Chen, Zhanfang. 2020. Effects of Feature Selection and Normalization on Network Intrusion Detection.

Umer, Muhammad Azmi, Junejo, Khurum Nazir, Jilani, Muhammad Taha, and Mathur, Aditya P. 2022. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection,* 100516.

Walfish, Steven. 2006. A review of statistical outlier methods. *Pharmaceutical technology* 30 (11): 82.

Wei, Yuanyuan, Jang-Jaccard, Julian, Sabrina, Fariza, and McIntosh, Timothy. 2019. MSD-kmeans: A novel algorithm for efficient detection of global and local outliers. *arXiv preprint arXiv:1910.06588.*

Wu, Renjie and Keogh, Eamonn. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering.*

Zhang, Ke, Hutter, Marcus, and Jin, Huidong. 2009. “A new local distance-based outlier detection approach for scattered real-world data.” In *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13,* 813–822. Springer.

Zhao, Yue, Nasrullah, Zain, and Li, Zheng. 2019. Pyod: A python toolbox for scalable outlier detection. *arXiv preprint arXiv:1901.01588.*