Anomaly detection for the Big Data

Xin Han

Xichen Tang

Shaowei Pan

# Introduction

Nowadays, data has increased in a large scale in various fields such as Internet, business, telecommunication, and biosciences. Big data is refered to describe the large and distributed nature of the data sets, this area has recently become a focus of scholarship. Doug Laney [26] defined challenges and opportunities brought about by increased data with a 3Vs model, i.e., Volume, Velocity, and Variety. Volume means the size of the data that becomes increasingly big with the collection of masses of data; Velocity means the timeliness of big data, specifically, data collection and analysis, etc. must be rapidly and timely conducted; Variety indicates the many types of data, which include structured, semi-structured, and unstructured such as video and text.

In recent years, the core challenges of big data have been extended to 4Vs which contained Volume, Variety, Velocity, and Value [15]. The Value refers to the benefits associated with the analysis of data and it highlights the meaning and necessity of big data, i.e., discovering the huge hidden values from datasets. Discovering abnormal patterns deviating from the datasets is refered as anomaly detection or outlier detection is being It can be used to detect anomaliers or events such as intrusion detection [16], fault detection [20], fraud detection [6], event detection in social networks [34].

An anomalier defined by Hawkins [18] as “an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.” Based on the definiton of Hawkins, two general approaches exist for the anomaly detection: rule-based method, where manually defining some rules of well-known anomaly behavior with prior knowledge, and data-based anomaly detection, where looking for behavior that is out of the normal from data. The Rule-based method works reliably on known anomaly behavior, but has the obvious disadvantage of not being capable of detecting new anomaly behavior, especially is not suitable in the age of big data. The data-based anomaly detection model works by identifying anomaliers by creating a model of the normal patterns in the data, and then compute an outlier score of a given data point on the basis of the deviations from these patterns [9]. The main advantage of data-based anomaly detection is that it does not require prior knowledge of an intrusion and thus can detect new anomaly behavior.

Because there is no rigid definition of which observation exactly is an anomalier, every method relying on certain assumptions of what qualifies as an anomalier. Some popular models are based on the distribution of objects [9, 24, 37], the distance [21] between objects, or on the density of the neighborhood of an object [1, 8, 30], or based on the ensemble method [43]. These methods represent different attempts to make the rather vague intuition about what anomalier are more concrete, typically in an implicit, procedural way.

This paper aims to review data-based anomaly detection methods for Big Data. The specially challenges of anomaly detection in the age of big data will be discussed and show how to reduce those challenges.

The rest of the chapter is organised as follows. We first introduce the existing methods of classic anomaly detection, in the Section [[sec-method]](#sec-method), Then, we introduce the explaining anomaly detect method in the Section [[sec-explain\_anomaly]](#sec-explain_anomaly). In the Section [[sec:evaluate]](#sec:evaluate), we introduce the popular evaluation method in anomaly detection. Section [[sec:tools]](#sec:tools) introduce the published tools of methods of anomaly detection, followed by the conclusions in the last section.

# Traditional Method of Anomaly Detection

[sec-method]

Many data-based anomaly detection methods have been proposed in data mining literature. The general methods can be divided into four categories as follows: statistical, distance, clustering and Ensemble [10].

## Statistical Based Methods

The statistical based detection assumed that if the difference between the data and the statistical distribution or the specified model is greater than a specific range [9], the data object is considered as anomalier. It include two special method: distribution-based approach and depth-based approach [40].

As for the distribution-based method, after given a distribution, a method of consistency checking is used to find anomalier. However, the actual distribution of the data set is always unknown, and it is difficult to estimate the data distribution in high-dimensional. To solve those problem, self-organizing map (SOM) [37], Support Vector Regression [24] (SVR) and other machine learning based methods are introduced to improve those shortcomings.

The depth-based approach takes into account that each object is a point with a specified depth in n-dimensional space, and that a data object may be anomalier with a lower depth. Although some dimension reduction methods such as Primary Component Analysis (PCA) [11], and Independent Component Analysis (ICA) are commonly used in this category, it still has a high computational complexity and has a low efficiency in the big data with high-dimensional and large data sets.

## Distance Based Methods

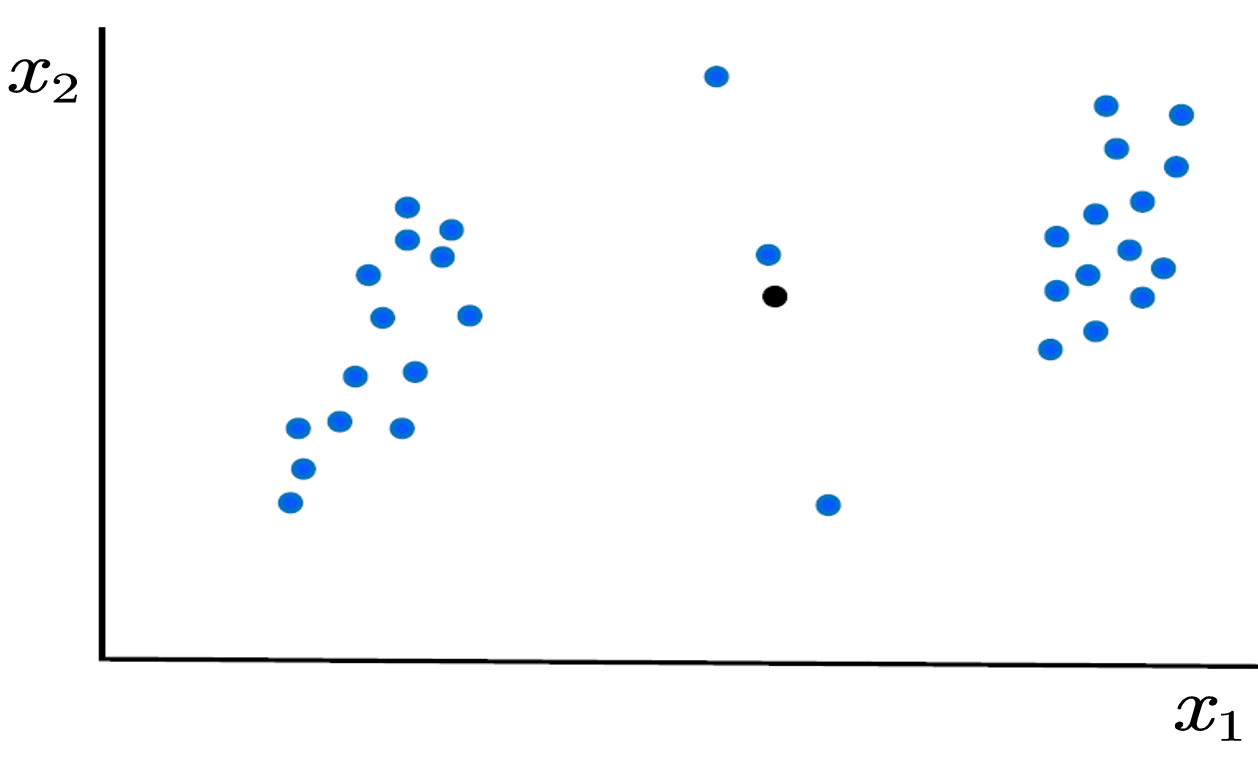


Figure 1: Distance Based Anomaly detection Methods

The distance based anomaly detection assumes that anomaliers are far away from other points. It calculate the distance between data points in data space after setting a distance function. A data object is regarded as anomalier when the distance between itself and others is large. The firstly anomaly detection method based on distance is proposed in [21]. Then it is extended to using K-neighbor distance to build the anomaly detector [25, 32]. The K-neighbor distance of each object is calculated and sorted from small to big, the objects which have largest distance are considered as anomaliers.

The distance based anomaly detection method is easier to realize, and is widely studied. But the complexity of the algorithm is relatively high, such as the computation complexity of KNN is , where the is the number of data points. and it cannot consider the size of data sets and the scalability of data dimension. Thus the practical application of distance based methods is limited.

## Density Based Methods

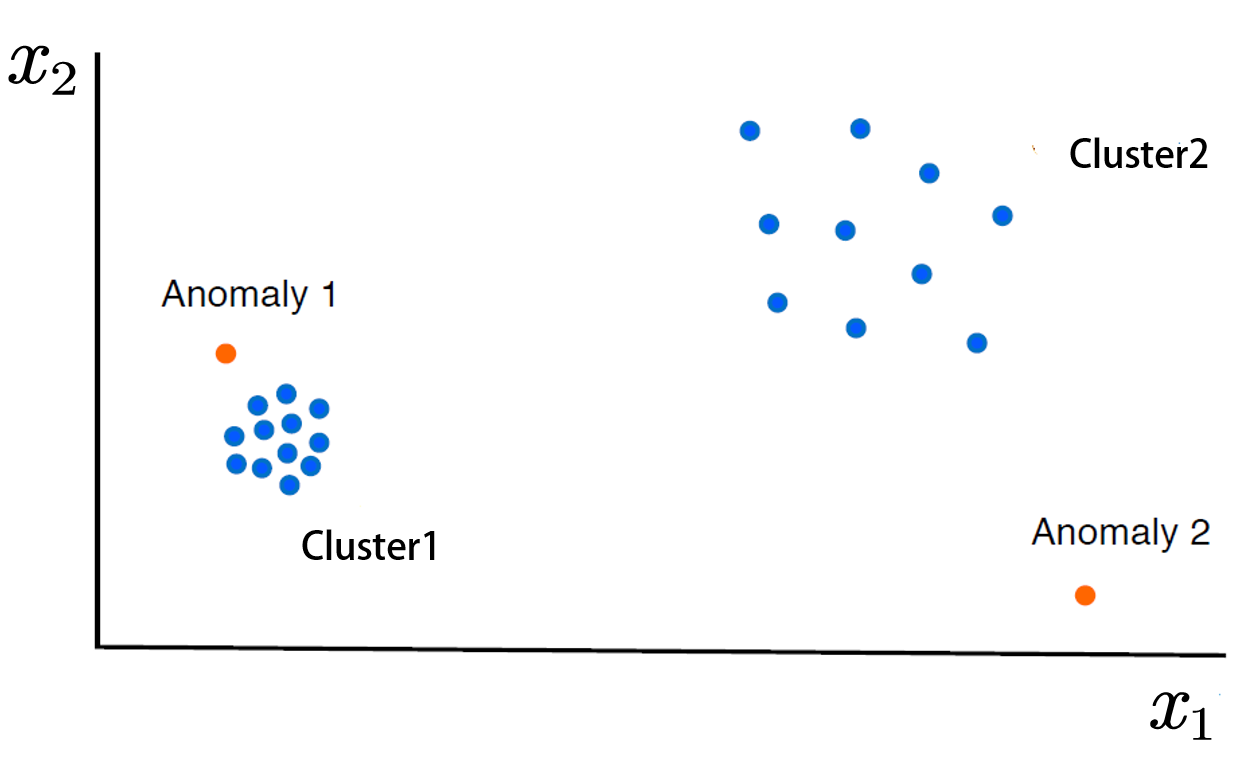


Figure 2: Density Based Anomaly detection Methods

Anomaliers are usually detected from an individual’s point of view, i.e. the anomaliers are far from their neighboring clusters. Therefore, it is not appropriate to use the overall distance as statistical based and distance based methods do. The density-based anomaly detection algorithms are proposed to solve this problem. The Local Outlier Factor (LOF) [8] is proposed to detect anomaliers by comparing the local density of an object to the local density of its neighbors. If the density of the data object is much lower than that of its neighbors, this data object is considered as anomalier. Local parsimony factor (LCF) is proposed to reduce the complexity of the computation by considering the maximum distance between the nearest objects as distances [1]. In addition, multi-granularity deviation factor (MDEF) is used as a measure of anomaly which comparing the number of data objects in neighbors and their mean [30]. It does not need to calculate the density of the data points, and the efficiency of the calculation is greater than LOF.

The density-based idea is closer to Hawkins’ definition of exception than the distance-based idea [18]. Therefore, it can detect local anomaliers, and have high detection accuracy. However, the time complexity is still very high, and the result of detection is sensitive to the selection of parameters such as the threshold of the outlier, which is difficult to determine.

## Ensembles Based Methods

Because each model is specifically designed for different characteristics of perception. It is only applicable to certain aspects of the “whole truth.” So it is best to integrate different third-party observational results to reach consensus. The main idea of this method (called “ensemble”) is that if these judgments do not contain all the same errors. It is useful to combine individual judgments or the results of marginalized observations [43]. One would think that this is a majority vote of the jury: one or another judgment on the statement may be wrong, but the majority judgment is still correct as long as the judgment is generally reliable. Isolation Forest [27] can be treated as typical ensemble method, it builds an ensemble of “Isolation Trees” (iTrees) for the data set, and anomaliers are the points that have shorter average path lengths on the iTrees. It has a low linear time complexity and a small memory requirement and is able to deal with high dimensional data with irrelevant attributes [9]. FuseAD [29] takes advantage of both statistical approach ARIMA and deep-learning-based approach CNN to propose an novel network: it shows great improvement on Yahoo Webscope benchmark.

# Challenges of Anomaly detection in big data

The two features of big data that have the greatest effect on the problem of high dimensionality are “volume” and “velocity.” brings additional challenges when data are increasing and arriving at speed as unbounded data streams. However, these strategies become highly complex and ineffective in detecting anomalies due to the challenges associated with large data sets and data streams. each problem has their individual challenges. Here, we present challenges brought by anomaly detection in big data. We have described each one of them in the following subsections.

## Challenges in the Volume of big data

In the age of big data, the design of anomaly detection methods have become increasingly complex. When the potential probability distribution is not known and the size of the data set is huge, the computational requirements increase. The Volume feature of big data emphasizes storage, memory, and computing power of the system to cope with ever-increasing data size [14]. When the data size is large, the traditional anomaly detection methods may become invalid, because of the limited computational power and associated factors. To overcome this issue, several parallel and distributed computing methods are proposed.

Managing computational power and disk input/output (I/O) communication can improve the efficiency of the method. D-cube [35] is a disk-based detection method to find fraudulent lockstep behaviour in large scale data and runs in a distributed manner across multiple machines. It is proved that could successfully detect network attacks and synchronised behaviour in rating data with highest accuracy. Nested loop (NL)[22] is a straightforward method to detect anomaliers in a database. Hung and Cheung [19] introduced an efficient and parallel version of the NL algorithm that reduces both computation and disk I/O costs.

Ramaswamy et al. [32] proposed a distance based anomaly detection method to detect anomalier in huge data sets. It segregates input data into separate subsets and prunes partitions that do not contain anomaliers, resulting in considerable savings in computation. DOLPHIN  [4] is also a distance based anomaly detection method that works on disk-resident data sets in huge datasets.

To solve the sequential exception problem in large databases, Arning et al. [5] proposed a linear algorithm using a dissimilarity function to capture the similarity rate of a data point. Erfani et al. [13] introduced an unsupervised method for high-dimensional large-scale unlabeled data sets to detect anomaliers that are a combination of a deep belief network (DBN) and one-class support vector machines (SVM). One-class SVMs (1SVMs) are used for detecting anomaliers through unsupervised learning and aim to model the underlying distribution of data while not considering irrelevant attributes or anomaliers in the training records. Features derived from training samples are taken as input to train 1SVMs. Conversely, a DBN is a multiclass semi-supervised approach and dimensionality reduction tool. It uses multilayer generative models (non-linear manifold) that learn one layer of features at a time from unlabeled data.

## Challenges in the Velocity of big data

Most traditional anomaly detection methods assume that the data set is generated by an unknown but stationary probability distribution , the volume of data is finite and the entire dataset could be stored for analysis [36]. However, in the big data age, the data set would be an infinite set of data instances in which each instance is a set of values with an explicit or implicit time stamp and it is refered as data stream [33]. The data stream are unbounded sequences and the entry rate is continuously high, as the respective variations repeatedly change over time. The anomaly detection on data stream, as show in Figure [3](#fig:streamAnomaly), is a highly challenge tasks because of the unbounded volume of data, the high rate of data generation, and data be stored will run out of memory space [33].

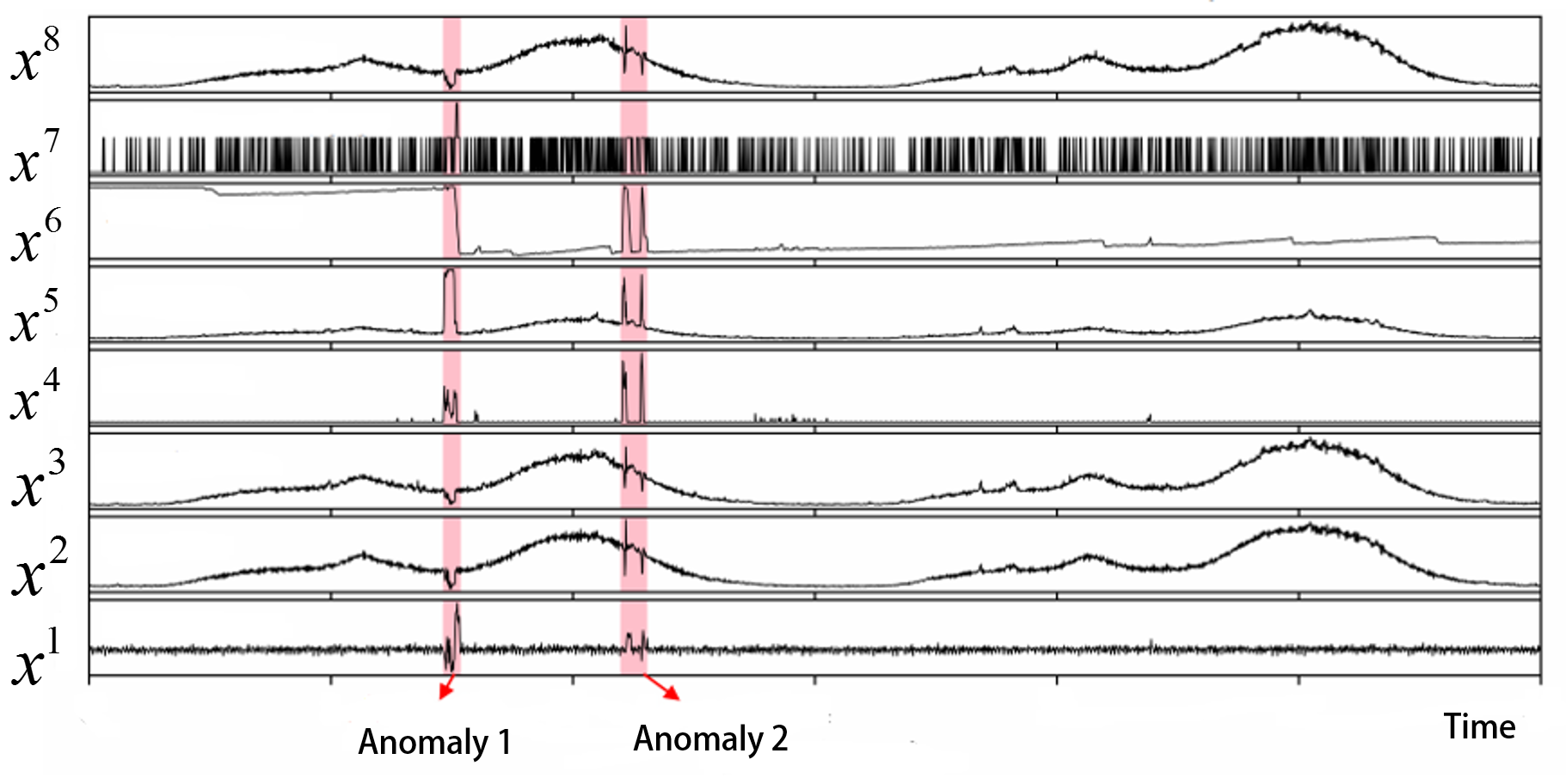


Figure 3: Anomaly detection on stream data

In most data stream scenarios， more recent information from the stream can reflect the emerging of new trends or changes on the data distribution. This information can be used to explain the evolution of the process under observation. In the sliding-window model， only the most recent information from the data stream are stored in a data structure whose size can be variable or fixed. This data structure is usually a first in， first out (FIFO) structure， which considers the objects from the current period of time up to a certain period in the past. The organization and manipulation of objects are based on the principles of queue processing, where the first object added to the queue will be the first one to be removed. In Figure [[slidingWindow]](#slidingWindow)， we present an example of the sliding-window model.

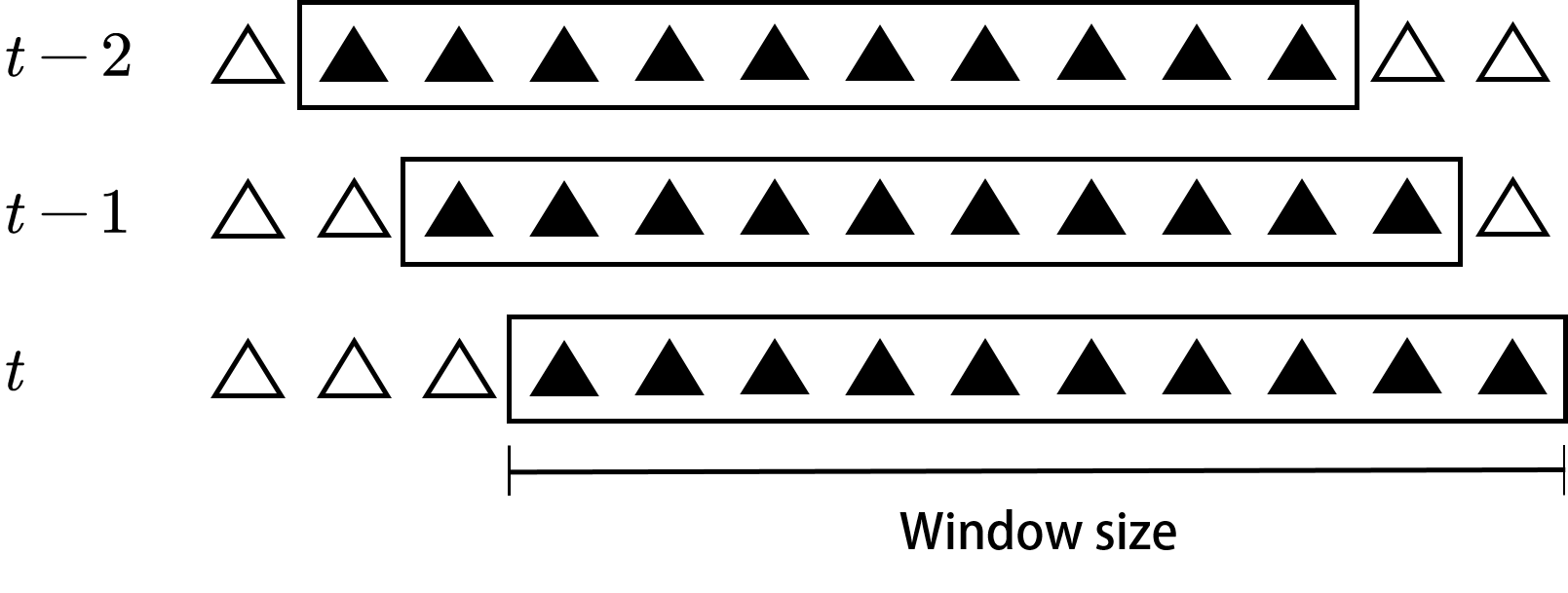


Figure 4: Sliding window model

Angiulli and Fasseti [3] introduced methods for recognizing distance-based anomaliers in data streams using a sliding window model in which anomaly queries are executed for the purpose of identifying anomaliers in the current window. Their algorithms execute anomaly queries and return an approximate answer based on accurate estimations with a statistical guarantee. These algorithms are based on a method known as stream outlier miner, or STORM, which is used to find anomaliers on distance based windowed data streams. A sliding window is used to continuously inspect the object until it expires. Later, Angiulli and Fasseti [2] presented an additional algorithm for identifying distance-based anomaliers in data streams using the sliding window model; although based on their previous approximate algorithm, this algorithm emphasized fixed memory requirements.

An algorithm based on the sliding window model for mining constrained frequent item sets on uncertain data streams was introduced by Yu et al. [41]. Known as CUSFgrowth (constrained uncertain data stream frequent item sets growth), the algorithm determines the order of items in transactions and analyzes the properties of constraints. According to the order of items determined by the properties of constraints a CUSF-tree is created; later, after the frequent item sets are satisfied, the constraints are mined from the CUSF-tree.

Kontaki et al. [23] also studied the problem of continuous anomaly detection in data streams using sliding windows. They proposed four algorithms that sought effective anomaly monitoring with lesser memory requirements. Apart from assuming that the data are in metric space, their model did not make any assumptions about the behavior of input data. Their methods have considerable flexibility with regard to parameter values, enabling the execution of multiple distance-based anomaly detection tasks with different values, and they reduced the number of distance computations using micro-clusters. Their primary concerns were to improve efficiency and reduce storage consumption. Their methods incorporate an event-based framework that bypasses unneeded computations benefiting from the expiration time of objects.

In addition, enseble-based methods are also proposed to resolve the problem of detecting anomaliers in streaming data. iForestASD [12], uses sliding window to deal with streaming data. On the current complete window, iForestASD uses the standard iForest method to to adapte streaming data anomaly detection. HS-Trees [38]

# Evaluation of Anomaly Detection

[sec:evaluate] Evaluation metrics are critical to building a successful anomaly detection system. Efforts have been made to determine the correct method for measuring the quality of abnormal detection. This section examines general evaluation metrics from two aspects: supervised detection and unsupervised detection which is distinguished by whether have labeled dataset.

## Supervised Evaluation

When the labeled dataset is valuable, the supervised evaluation method could be used. At present, the popular evaluation methods of supervised learning include accuracy and recall, F1 score, ROC curve, and AUC.

### Confusion Matrix

The confusion matrix, as shown in Table [1](#tb:confusion), is a table two dimensions where each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class [31]. TP means true positives (i.e. items correctly labeled as belonging to the positive class), FP means false positives (i.e. items incorrectly labeled as belonging to the positive class), FN means false negatives (i.e. items which were not labeled as belonging to the positive class but should have been). TN means true negative (i.e. items correctly labeled as belonging to the negative class).

Confusion Matrix. TP is True Positive; FP is False Positive; TN is True Negative; FN is False Negative.

|  |  |  |
| --- | --- | --- |
|  | Actual value | Actual value |
| Predicted value | TP | FP |
| Predicted value | FN | TN |

### Precision

The *Precision* measures the ratio of examples classied as positive that are truly positive [39].

### Recall

The *Recall* measures the ratio of positive examples that are correctly labeled.

### F1 Score

The *F1 score* is a tradeoff between the *Precision* and *Recall*:

F1 metric weights recall and precision equally, and a good detection algorithm will maximize both precision and recall simultaneously. Moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

### AUC

*AUC* (area under curve) refers to the area under the *ROC curve*. The vertical and horizontal axis ranges are (0,1), so the total area is less than 1 [7]. The larger the *AUC*, the better the classification effect.

The *AUC* value can be think as the probability that the two randomly selected objects (positive example (anomalier) and negative example (normalier)) are arranged correctly (i.e., anomalier is classified before the normalier value) [17]. The *ROC curve* and *AUC* analysis inherently address the problem of imbalance using relative frequencies, making them particularly popular in evaluating the detection of exclusion.

## Unsupervised Evaluation

If some task have no labeled dataset, the unsupervised evaluation method such as comparative evaluation, generating pseudo tags, and similarity analysis could be used.

### Comparative Evaluation

For unsupervised learning, a common evaluation strategy is to rank the results according to the score of anomaliers, and then iteratively set the threshold from the first to the last. This will form n ancestor values (true positive rate and false positive rate), and a ROC curve can be obtained. The integral AUC of ROC can be used as a measure to test the performance.

### Generating Pseudo Tags

There are a lot of learning efforts to transform unsupervised learning into supervised learning, and there are now feasible methods. Then, we can use the evaluation methods of supervised learning, such as accuracy.

### Similarity Analysis

Unsupervised learning often depends on the similarity between data, which can be expressed as spatial density or distance measurement. In the evaluation of anomaly detection algorithm, it can be assumed that a large number of normal data are closely adjacent (can form multiple clusters), and anomaliers are often quite different from these normal points.

# Anomaly detection software packages

[sec:tools] Many companies have build their own anomaly detection systems in order to meet their specific business needs. However, there are still many open source anomaly detection packages available. In this section, we review some of the popular software packages for practitioners to build their anomaly detection systems.

## PyOdds

PyODDS (<http://pyodds.com/>) is an end-to end Python system for anomaly detection. It provides several anomaly detection algorithms and support both static and time-series data type.

## PyOD

PyOD [42] (<https://pypi.org/project/pyod/>) is a comprehensive and scalable Python toolkit for detecting anomalier in multivariable data. It includes more than 20 detection algorithms, including new deep learning models and ensembles methods.

## ADTK

Anomaly Detection Toolkit (ADTK) (<https://adtk.readthedocs.io>) is a Python package for unsupervised / rule-based time series anomaly detection.

## Scikit-Multiflow

Scikit-Multiflow [28] is the main open source machine learning framework for multi-output, multi-label and data streaming. Implemented in Python language, it includes various algorithms and methods for streams mining and in particular the popular HS-Trees algorithm.

## Distributed Computing frameworks

Distributed computing is one of the most important techniques to handle and process big data. MapReduce is one of the first distributed programming paradigms to handle big data storing and processing. Henceforth, many frameworks for distributed computing such as Apache Hadoop (<https://hadoop.apache.org/>), Apache Storm (<https://storm.apache.org/>), Apache Spark (<https://spark.apache.org/>), Apache Flink (<https://flink.apache.org/>) were developed addressing the increasing demands of big data. Most of these frameworks have in-house machine learning (ML) libraries, and Apache Spark has a powerful ML library than any of the other frameworks.

# Conclusions

Anomoaly detection has been extensively attentted in recent years in the modern industry. This chapter aimed to present the state of the art anomaly detecttion for practical using in industry. This has included anomoaly detection method based on distribution, distance, density, cluster and ensemble. To evaluate anomoaly detection method, we reviewed commonly used two categories including supervised and unspervisd. In addition, we reviewd the explanation of anomoaly detection which has been getting more and more attention. Finaly, we provided a list of free and open source software packages for practitioners to create their own anomaly detection systems.

A wide range of studies have been carried out in the field of abnormal detection. As this chapter only coveres this topic, we recommend reader to read more detail in the references of this chapter.

# Reference

1. Agyemang, M.: Algorithm for mining local outliers. In: Innovations through information technology: 2004 information resources management association international conference, new orleans, louisiana, USA, may 23-26, 2004. p. 5 IGI Global (2004).

2. Angiulli, F. et al.: Detecting outlying properties of exceptional objects. Acm transactions on database systems (tods). 34, 1, 1–62 (2009).

3. Angiulli, F., Fassetti, F.: Detecting distance-based outliers in streams of data. In: Proceedings of the sixteenth ACM conference on conference on information and knowledge management. pp. 811–820 (2007).

4. Angiulli, F., Fassetti, F.: Very efficient mining of distance-based outliers. In: Proceedings of the sixteenth ACM conference on conference on information and knowledge management. pp. 791–800 (2007).

5. Arning, A. et al.: A linear method for deviation detection in large databases. In: KDD. pp. 972–981 (1996).

6. Bolton, R.J., Hand, D.J.: Statistical fraud detection: A review. Statistical science. 235–249 (2002).

7. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition. 30, 7, 1145–1159 (1997).

8. Breunig, M.M. et al.: LOF: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data. pp. 93–104 (2000).

9. Chandola, V. et al.: Anomaly detection: A survey. ACM computing surveys (CSUR). 41, 3, 1–58 (2009).

10. Cook, A. et al.: Anomaly detection for IoT time-series data: A survey. IEEE Internet of Things Journal. (2019).

11. Deng, X. et al.: Modified kernel principal component analysis based on local structure analysis and its application to nonlinear process fault diagnosis. Chemometrics and Intelligent Laboratory Systems. 127, 195–209 (2013).

12. Ding, Z., Fei, M.: An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. IFAC Proceedings Volumes. 46, 20, 12–17 (2013).

13. Erfani, S.M. et al.: High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition. 58, 121–134 (2016).

14. Gadepally, V., Kepner, J.: Big data dimensional analysis. In: 2014 IEEE high performance extreme computing conference (HPEC). pp. 1–6 IEEE (2014).

15. Gantz, J., Reinsel, D.: Extracting value from chaos. IDC iview. 1142, 2011, 1–12 (2011).

16. Garcia-Teodoro, P. et al.: Anomaly-based network intrusion detection: Techniques, systems and challenges. computers & security. 28, 1-2, 18–28 (2009).

17. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology. 143, 1, 29–36 (1982).

18. Hawkins, D.M.: Identification of outliers. Springer (1980).

19. Hung, E., Cheung, D.W.: Parallel mining of outliers in large database. Distributed and Parallel Databases. 12, 1, 5–26 (2002).

20. Hwang, I. et al.: A survey of fault detection, isolation, and reconfiguration methods. IEEE transactions on control systems technology. 18, 3, 636–653 (2009).

21. Knorr, E.M., Ng, R.T.: A unified notion of outliers: Properties and computation. In: KDD. pp. 219–222 (1997).

22. Knox, E.M., Ng, R.T.: Algorithms for mining distancebased outliers in large datasets. In: Proceedings of the international conference on very large data bases. pp. 392–403 Citeseer (1998).

23. Kontaki, M. et al.: Continuous monitoring of distance-based outliers over data streams. In: 2011 IEEE 27th international conference on data engineering. pp. 135–146 IEEE (2011).

24. Kromanis, R., Kripakaran, P.: Support vector regression for anomaly detection from measurement histories. Advanced Engineering Informatics. 27, 4, 486–495 (2013).

25. Kuang, L., Zulkernine, M.: An anomaly intrusion detection method using the CSI-KNN algorithm. In: Proceedings of the 2008 ACM symposium on applied computing. pp. 921–926 (2008).

26. Laney, D.: 3D data management: Controlling data volume, velocity, and variety. META Group (2001).

27. Liu, F.T. et al.: Isolation forest. In: 2008 eighth IEEE international conference on data mining. pp. 413–422 IEEE (2008).

28. Montiel, J. et al.: Scikit-multiflow: A multi-output streaming framework. The Journal of Machine Learning Research. 19, 1, 2915–2914 (2018).

29. Munir, M. et al.: FuseAD: Unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models. Sensors. 19, 11, 2451 (2019).

30. Papadimitriou, S. et al.: Loci: Fast outlier detection using the local correlation integral. In: Proceedings 19th international conference on data engineering (cat. No. 03CH37405). pp. 315–326 IEEE (2003).

31. Powers, D.M.: Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation. (2011).

32. Ramaswamy, S. et al.: Efficient algorithms for mining outliers from large data sets. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data. pp. 427–438 (2000).

33. Sadik, S., Gruenwald, L.: Research issues in outlier detection for data streams. Acm Sigkdd Explorations Newsletter. 15, 1, 33–40 (2014).

34. Sakaki, T. et al.: Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th international conference on world wide web. pp. 851–860 (2010).

35. Shin, K. et al.: D-cube: Dense-block detection in terabyte-scale tensors. In: Proceedings of the tenth ACM international conference on web search and data mining. pp. 681–689 (2017).

36. Silva, J.A. et al.: Data stream clustering: A survey. ACM Computing Surveys (CSUR). 46, 1, 1–31 (2013).

37. Siripanadorn, S. et al.: Anomaly detection in wireless sensor networks using self-organizing map and wavelets. International Journal of Communications. 4, 3, 74–83 (2010).

38. Tan, S.C. et al.: Fast anomaly detection for streaming data. In: Proceedings of the twenty-second international joint conference on artificial intelligence - volume volume two. pp. 1511–1516 AAAI Press, Barcelona, Catalonia, Spain (2011).

39. Ting, K.M.: Precision and recall. Encyclopedia of machine learning. 781, (2010).

40. Wu, H.-S.: A survey of research on anomaly detection for time series. In: 2016 13th international computer conference on wavelet active media technology and information processing (ICCWAMTIP). pp. 426–431 IEEE (2016).

41. Yu, Q. et al.: Uncertain frequent itemsets mining algorithm on data streams with constraints. In: International conference on intelligent data engineering and automated learning. pp. 192–201 Springer (2016).

42. Zhao, Y. et al.: PyOD: A python toolbox for scalable outlier detection. Journal of Machine Learning Research. 20, 96, 1–7 (2019).

43. Zhou, Z.-H.: Ensemble methods: Foundations and algorithms. CRC press (2012).