# Application of Outlier Detection Methods in Audit Analytics

Kevin Fulcer
Baker Tilly US, LLP

Hanchi Gu
Shanghai University of Finance and Economics

Hanxin Hu
Kean University

Qing Huang
Marshall University

Alexander Kogan
Rutgers University

Miklos A. Vasarhelyi
Rutgers University

Danyang Wei
Durham University

Jimmy Young
AICPA

Running Head: Application of Outlier Detection Methods in Audit Analytics

**Acknowledgment:** The authors express their sincere gratitude to Phillip McCollough for his numerous critical contributions to this research. The author names are listed in alphabetical order by last name. Hanchi Gu is supported by the MOE Project of Key Research Institute of Humanities and Social Science at University [No.22JJD790093], the 111 Project (B18033), and the Fundamental Research Funds for the Central Universities.

The opinions expressed are those of the authors and do not represent the opinions or positions of AICPA and our industry partners.

Alexander Kogan and Miklos A. Vasarhelyi, Rutgers University, Rutgers Business School, Accounting & Information Systems Department, Newark, New Jersey, USA; Kevin Fulcer, Baker Tilly US, LLP, Assurance Operations, Madison, Wisconsin, USA; Hanchi Gu, Shanghai University of Finance and Economics, Institute of Accounting and Finance, Shanghai, China; Hanxin Hu, Kean University, Accounting and Finance Department, Union, New Jersey, USA; Qing Huang, Marshall University, Brad D. Smith Schools of Business, Department of Marketing, Management Information Systems, & Entrepreneurship, Huntington, West Virginia, USA; Danyang Wei, Durham University, Business School, Department of Accounting, Durham, United Kingdom; Jimmy Young, Association of International Certified Public Accountants (AICPA), Accounting and Auditing Innovation Department, St. Louis, Missouri, USA.

**JEL Classification:** M42, M40, C45

**Keywords:** Audit Analytics, Outlier Detection, Unsupervised Learning, Credible Algorithms

# Application of Outlier Detection Methods in Audit Data Analytics


## ABSTRACT

Audit transaction anomalies can be viewed as outliers. Unsupervised learning methods of outlier detection do not require outcome labels and enable auditors to discover possible problems based on observed transaction patterns. This study develops a framework for using outlier detection methods in audit selection and evaluates the proposed framework on real-world revenue sub-ledger datasets. The results indicate that the proposed framework could facilitate the identification of relevant outlier detection algorithms and effectively select risky observations.


## I. MOTIVATION FOR APPLYING OUTLIER DETECTION METHODS

"Outliers" refer to data points that significantly deviate from other observations, leading to the suspicion that they were generated by different mechanisms (Hawkins 1980). Methods for identifying outliers were proposed first in the 1980s (Denning 1987). Auditing anomalies can be viewed as outliers. Identifying transaction anomalies is challenging for auditors, particularly due to the emergence of big data and high transaction volume, which makes it extremely difficult to manually select the most anomalous transactions for examination.

Recent decades have seen the development of advanced outlier detection techniques, which are now utilized in various domains such as finance, insurance claims, credit card fraud, healthcare services, loan processing, and network security (Thiprungsri and Vasarhelyi 2011; Malini and Pushpa 2017; van Capelleveen et al., 2016; Hodge and Austin 2004; Jabez and Muthukumar 2015). While these methods may be useful in various stages of audit engagements such as risk identification and assessment, this research focuses on their application in substantive testing. We argue that outlier detection methods can be a useful addition to audit data analytics (ADA) for

identifying higher-risk observations for subsequent examination. ADA can be applied through the discovery and analysis of patterns, identification of anomalies, and extraction of other useful information from data related to the audit subject matter (AICPA 2015) (see Appendix A for the definitions of analytical procedures in auditing standards). This research focuses on the application of unsupervised outlier detection methods for selecting transactions for manual examination.

Based on how professional expertise and client knowledge are used, knowledge-based methods in transaction selection can be categorized as:

- Expert system type methods (see e.g., No, Lee, Huang, and Li 2019), in which experts formalize their professional expertise and client knowledge as risk filters that are used to evaluate observations. These filters incorporate various risk factors and need to be customized for different clients and scenarios. Ensuring the filters are effective requires in-depth analysis and testing. As clients and risks evolve, the filters need updates, necessitating significant ongoing effort.
- Supervised learning methods (see e.g., Bao, Ke, Li, Yu, and Zhang 2020), which derive evaluations of current observations after training a machine learning model on a set of archival observations that have outcome labels indicating potential risk. Acquiring these labels can be very costly or even impossible since the size of the training dataset needs to be considerable.
- Unsupervised learning methods, such as outlier detection, identify observations that differ significantly from the overall dataset, without relying on any outcome labels. Outlier detection methods process observation attributes to generate outlier scores that rank observations by the likelihood of being abnormal (various methods are summarized in Table 2 in Appendix B). When employing outlier detection methods, auditors conduct

follow-up investigations of observations that have the highest outlier scores. Using these methods can significantly reduce the auditors' burden of labeling a large set of transactions as risky or not risky (a comparison of supervised and unsupervised methods is summarized in *Appendix* D). It is also much less burdensome than creating a comprehensive set of risk filters.

Due to the advantages of outlier detection methods outlined above, it is important to develop an ADA framework for utilizing these methods in audit selection. The development of such a framework is the objective of this paper. This framework is proposed to complement other ADA methodologies based on knowledge engineering and supervised learning in obtaining sufficient audit evidence.

Different outlier detection methods calculate outlier scores based on different mathematical approaches to derive their models of observation regularity. It is possible that a particular approach may not be suitable for a particular dataset. Therefore, outlier detection results may be irrelevant with respect to specific audit objectives. To address this issue, we propose a two-stage framework that could identify meaningful and relevant suspicious transactions in a full population. In the initial run, it is determined which algorithms are "credible," i.e., the most aligned with audit objectives. The determination of alignment is based on how well these algorithms identify the so-called "obvious outliers," i.e., a very small set of highly risky transactions chosen by the auditors. Then, in the second run, the selected credible algorithms are applied to the dataset with the obvious outliers removed, and the outlier scores are combined to determine the outliers to be selected for further examination. Notably, these selected outliers can potentially include important risky transactions not suspected by human auditors. The designed two-stage framework constructs a "weakly" supervised method (explained in Appendix E).

This paper contributes to the ADA literature in three ways.

- To the best of our knowledge, this is the first research to propose and validate an outlier detection framework that incorporates the choice of credible algorithms and assists auditors in identifying non-obvious risky observations with minimal effort (compared with other ADA approaches).
- The proposed framework could mitigate the bias induced by algorithm selection. In contrast to previous research that relies on one or two outlier detection algorithms to identify risky observations (e.g., Thiprungsri and Vasarhelyi 2011), the framework considers an array of credible algorithms and ranks the anomalousness of observations using ensemble outlier scores.
- The proposed framework utilizes a credibility test to identify those outlier detection algorithms that are most consistent with auditors' judgments. It can reduce the concern that the unsupervised nature of outlier detection algorithms may impair the outlier scores' relevance to specific audit objectives.

The remaining sections are organized as follows. Section II identifies the most prevalent outlier detection methods in the literature, illustrating their differences. Section III outlines the challenges of implementing outlier detection methods in audits. Using a procurement dataset, Section IV develops a design artifact, a framework for applying outlier detection methods in audits. Following Design Science research guidelines, Section V employs the case study method to evaluate the validity of the proposed framework on revenue sub-ledger datasets. Section VI summarizes the proposed framework and discusses research limitations.

## II. OVERVIEW OF POPULAR OUTLIER DETECTION METHODS

It is reasonable to expect that outlier detection methods successfully applied in various domains can be promising candidates for ADA applications. Thus, the development of a framework for using outlier detection methods in audit selection should start with analyzing outlier detection literature. To identify the most prevalent categories of outlier detection methods in the extant literature, we employed the software program *Publish or Perish* to find highly cited articles on outlier detection. General search terms included "outlier detection review," "outlier detection survey," "anomaly detection survey," etc. The results from 2003 to 2022 identify 23 detailed outlier detection survey articles (Table 1 in Appendix B exhibits the summary of search results). Following an examination of these articles, we identified six major types of outlier detection methods: Statistics-based, Distance-based, Density-based, Clustering-based, Deep learning-based, and Ensemble-based. Table 2 in Appendix B describes the rationale and representative methods for each type, and how the various methods differ.

It is important that these methods, in addition to generating binary classification outcomes (i.e., abnormal versus normal designation), provide users with outlier scores indicating the degree to which an algorithm considers the points to be outliers (Sejr and Schneider-Kamp 2021). Usually, the scores generated by these algorithms vary significantly. Figure 1 illustrates outlier scores calculated on the same Mall-Customer-Segmentation dataset using LOF, KNN, and HBOS algorithms. Often HBOS assigns higher outlier scores to data points located in central regions, whereas KNN and LOF usually consider data points positioned in boundary regions to be outliers. Therefore, caution should be exercised when choosing outlier detection methods.

Most outlier detection methods are parameterized and thus not fully specified as algorithms without choosing parameter values. We refer to outlier detection methods with specified parameter values (e.g., 5NN) as outlier detection algorithms.

## III. CHALLENGES

Given the considerable effort demanded from auditors by expert systems and supervised learning type methods, this study focuses on the application of unsupervised methods to improve the selection of risky transactions for examination. Although outlier detection has the potential to identify higher-risk items, its application by auditors presents certain challenges.

### Algorithm Selection

Bias in machine learning refers to a set of assumptions an algorithm makes about the properties of a dataset (Alpaydin 2020). The performance of an algorithm depends on how well its bias fits the actual data. Different algorithms are specialized for different characteristics of datasets (Zimek, Campello, and Sander 2014). Auditors face challenges in choosing the most suitable algorithm. For some methods, the results are greatly affected by parameters (Breunig, Kriegel, Ng, and Sander 2000), making it difficult to decide which parameters yield the most useful results.

### Understandability

In outlier detection, the mechanism of a method is described in both abstract concepts and concrete programmed functions. This formal knowledge does not equate to the understanding of the results if the method is complex.

KNN, one of the simpler outlier detection methods, is used to illustrate this challenge. The mechanism of KNN is based on calculating the distance of observation to its $k$ nearest neighbors (Ramaswamy, Rastogi, and K. Shim 2000) and identifying observations having a large distance as outliers. Although the mechanism is clearly defined, it is complicated to understand which

attributes contribute to observations being outliers, making it challenging for auditors to develop an appropriate response.

**Evaluation**

Direct evaluations of the results of outlier detection algorithms using cross-validation (explained in Appendix G) are usually impossible due to the lack of labels on observations to compare against.

To verify the results of outlier detection algorithms, auditors need to conduct manual examinations of observations to establish their true nature. Outlier detection algorithms may produce two types of errors. False positive errors are those observations that the algorithm identifies as abnormal, but the auditors would not consider to be risky. False negative errors are observations that the algorithm identifies as normal, but the auditors would consider risky. It is usually impractical to calculate the total error rate of an algorithm since the costs associated with these errors can differ significantly. Outlier detection methods construct models of normalcy assuming that the vast majority of observations in the dataset are normal and the number of abnormal observations is proportionally small. Otherwise, it is inappropriate to use them. Given a small number of identified abnormal observations, it is feasible to evaluate their riskiness and thus determine the false positive errors. However, in larger datasets (which are not uncommon), it is not feasible to evaluate the riskiness of all observations that are categorized as normal. Therefore, it is not practical to determine the false negative errors.

## IV. FRAMEWORK

We propose a framework (see Figure 2) for auditors to use outlier detection algorithms in identifying higher-risk observations for further examination. The framework incorporates data

preprocessing, algorithm selection, and execution of the outlier detection algorithms twice. The first execution is for selecting algorithms that auditors would deem credible concerning the audit objectives. The second execution uses these credible algorithms to construct an ensemble aimed at selecting riskier observations for further investigation. The details of the preprocessing steps of the framework are summarized in Appendix C.

**Algorithm Selection**

Selecting the appropriate outlier detection algorithm is often challenging. To define an algorithm, auditors need to specify the method and parameter values. The number and range of parameters differ for outlier detection methods. It is often difficult to know which parameter values are optimal for achieving the objectives. We recommend that auditors consider various outlier detection methods. For each method, different parameter values should be examined (e.g., Pillai, Raghuwanshi, and Gaikwad, 2020; Ramasamy, Kadry, and Lim, 2021). Data considerations for algorithm selection are briefly described in Table 4 of Appendix B.

**Obvious Outliers**

As an alternative to understanding the mechanics of outlier detection methods, auditors can evaluate the relevance of the results they produce.

We propose a framework to establish the credibility of outlier detection algorithms by assessing their ability to identify what auditors would consider "obvious outliers." These obvious outliers should be defined by auditors based on the audit objectives. To pinpoint obvious outliers, auditors might select a small subset of relevant attributes and choose items with extreme values in those attributes. However, many methods for identifying obvious outliers are possible. Further research

is needed on the optimal method of selecting obvious outliers. It is expected that credible algorithms will flag as abnormal those transactions that present risks relevant to the audit objectives.

Additionally, the credibility of outlier detection algorithms can be evaluated based on their success in identifying those observations that have already been examined. This method of establishing credibility exploits the fact that various audit procedures may have been completed before employing outlier detection methods. Therefore, outcomes of already executed audit procedures may indicate a pattern of abnormal and normal observations.

**Evaluating Algorithm Results**

Computational procedures in outlier detection utilize every observation in the dataset to derive a model of normalcy and to evaluate how well an observation fits into that model by calculating its outlier score. This score will determine whether an observation is viewed as normal or abnormal. A way to make an equitable comparison of the results of outlier detection algorithms as well as conventional audit sampling approaches is to limit the number of selected (highest risk) observations to the same value.

The best-performing outlier detection algorithms are the algorithms that generate the fewest number of false positives and the fewest number of false negatives. Since the number of highest risk observations (i.e., the selection size) is chosen to be the same for every outlier detection algorithm, it is possible to identify the best-performing algorithm based on the fewest false positives. This is the case because the minimum number of false positives implies the maximum number of true positives, and the maximum number of true positives implies the minimum number of false negatives since the sum of true positives and false negatives equals the number of risky

observations in the dataset. Accordingly, the concept of P@n (Craswell 2009) in information retrieval, where Precision at n is the proportion of the top-n ranked documents that are relevant, can be used as the metric for algorithm evaluation. If n is the selection size, then P@n is the percentage of verified risky items in the top n ranks of items based on their outlier scores:

$$\text{P@n} = \frac{R}{n},$$

where n is the number of top-ranked abnormal items and R is the number of true outliers among them (see Appendix F for more details). A comparison among different algorithms can be made based on their P@n.

**Initial Run**

The objective is to establish the credibility of outlier detection algorithms. Auditors can use outlier detection algorithms with various parameter settings for each method and select those algorithms that have better performance. The performance metric is the number of obvious outliers among the abnormal observations selected. Auditors can choose the minimum value of the performance metric as the credibility threshold for choosing credible algorithms based on the initial run. For example, if 100 is the selected size, 50 is the total number of obvious outliers, and 30 is the credibility threshold, any algorithm that successfully identifies 30 or more obvious outliers in its 100 top-ranked observations would be credible. The initial run enables choosing those algorithms that exhibit similarity to auditors' judgment in identifying the obvious outliers. Therefore, the chosen credible algorithms are expected to select abnormal observations in alignment with the audit objectives. This supports relying on the results of credible algorithms, though it may come at the cost of a reduced likelihood of detecting unknown abnormalities. It is possible that none or

multiple of the outlier detection algorithms may be credible. If none is credible, the auditors should not proceed with using outlier detection algorithms.

**Obvious Outliers Removal**

Next**,** the obvious outliers should be removed from the dataset before proceeding. Not removing the obvious outliers is potentially harmful as they increase the contamination rate[1] of the dataset, biasing the constructed models of normalcy in the second run. Since the number of obvious outliers (provided by the auditors) is small, the difference between the results of the initial and the second run is likely to become less significant as the size of the dataset increases.

**Second Run**

The second run is to determine what other observations appear abnormal and require additional attention. All credible algorithms will be applied to the dataset (after removing the obvious outliers) to identify unknown outliers. With multiple credible algorithms, second-run results will likely differ by algorithm. Auditors can combine these results to make the selection decision (i.e., the selection decision is based on the results of an ensemble of credible algorithms). A simple approach to creating an ensemble (which is used herein) is to aggregate the outlier scores of all credible algorithms with equal weights (i.e., calculate the mean score). Another approach may be to aggregate the outlier scores of the credible algorithms separately for each outlier detection method and calculate the final score as the average of the methods. Regardless of the approach

[1] The contamination rate refers to the proportion of outliers to the total number of observations in the dataset.

taken, auditors need to choose the selection size and use the results of the ensemble to select observations for further investigation.

## V. EVALUATION

This research follows the Design Science guidelines proposed by Hevner, March, Park, and Ram (2004). In this study, the proposed design artifact is the framework outlined in Section IV. As discussed by Hevner et al. (2004) and Simon (1996), the nature of the design process is an iterative Generate/Test Cycle. An artifact is designed to solve a problem. The performance of the newly created artifact is then tested. Feedback is gathered based on the test results, and is used to modify the artifact, leading to another round of generation and testing. The framework is first developed and refined using a procurement dataset (see Appendix H). Then, it is tested by applying it to other datasets. Under the Design Science Research Guideline 3 (design evaluation), we use a case study to investigate the proposed artifact and demonstrate its potential.

A revenue sub-ledger dataset provided by practicing auditors is used for the framework evaluation (see Table 5 in Appendix H). Eight prevalent outlier detection methods (briefly described in Table 3 of Appendix B) are used for the initial run on the preprocessed revenue dataset. Table 1 in Appendix B summarizes the parameter configurations utilized by each method (265 combinations of parameter settings considered). Prior to the initial run, the auditor-collaborator executed traditional audit procedures (e.g., to determine extreme values of fundamental attributes) and identified 48 obvious outliers.

In the initial run, we choose 100 as the selection size and calculate Precision at 100, i.e. the number of obvious outliers identified by each algorithm. The MCD algorithms exhibited the maximum

degree of "similarity" to auditor judgment, with some parameter settings (e.g., support_fraction = 0.55) identifying 21 obvious outliers in 100 top-ranked observations. HBOS algorithms with varying parameter configurations did not identify any obvious outliers, suggesting the outlier detection mechanism underlying this method is incompatible with auditor judgment on this dataset. Using a loose (stringent) credibility threshold of 9 (18), we choose 32 (3) outlier detection algorithms as exceeding the credibility threshold and apply them to the dataset after removing the obvious outliers.

Based on the second run results, we compute the aggregation of normalized outlier scores generated by 32 (3) outlier detection algorithms exceeding the loose (stringent) credibility threshold, and rank observations based on their ensemble outlier scores. To evaluate the efficacy of credible algorithms in identifying non-obvious observation outliers, our auditor-collaborator examined the 25 top-ranked observations selected by both loose and stringent credible algorithm ensembles and concluded that both selections are consistent with the objectives of the audit procedures, and observations flagged by credible algorithms as outliers are indeed higher-risk observations that warrant further investigation. In both loose and stringent cases, four observations with a certain vendor "X" are always selected. The auditors concluded that these transactions may not be selected using traditional audit methods and that these transactions justify follow-up. The auditors viewed the stringent selection as more relevant, implying that more credible outlier detection algorithms generate more relevant selection.

In addition to the evaluation described above that utilized archival data, we evaluated the proposed framework during an active audit engagement. We obtained a revenue dataset for the fiscal year ending in the fall of 2023 (see Table 5 in Appendix H). Our practitioner partners selected 32 obvious outliers. Using a loose (stringent) credibility threshold of 9 (12), 33 (10) outlier detection

algorithms exceeding the credibility threshold were chosen and then applied to the dataset after removing the obvious outliers. The auditors tested the top 30 selections of each ensemble (with 20 of them overlapping, the number of selected unique transactions was 40) during fieldwork. Out of these 40 transactions, one lacked any supporting documents, and four transactions did not have any corresponding payments at the time. After two weeks of follow-up investigation, the transaction without supporting documentation remained in this status and was found to be processed not in compliance with the business rules. One transaction remained without corresponding payments, even though the transaction occurred well before the end of the fiscal year. The auditors judged the discovery of the transaction without supporting documentation as an important and unusual finding for the client. In this engagement, the auditors also viewed the stringent selection as more effective than the loose selection method, further suggesting that more credible outlier detection algorithms generate more relevant selection.

## VI. CONCLUDING REMARKS

This research develops a framework for the application of outlier detection techniques in audits. The proposed two-stage framework can facilitate the identification of relevant outlier detection algorithms and effectively identify risky observations. The framework offers a method to verify a model's credibility and detect outliers without significant data labeling effort. The evaluation case study demonstrates the success of the proposed framework in identifying transactions for follow-up. This framework identifies outliers likely overlooked by other procedures. While Table 2 of Appendix B presents currently popular outlier detection methods, the proposed framework is designed to be capable of integrating any future outlier detection methods for improved

performance, as more refined methods become available. The proposed framework should be used together with other audit procedures, as it is intended to supplement, not replace them.

This research has limitations. Auditors may not know which attributes are causing certain observations to be selected for further examination, which may impede the implementation of the framework in practice. Further, we rely on the obvious outliers identified by an auditor to establish the credibility of algorithms. Auditors may choose varying obvious outliers due to the judgmental nature of selection. Changing the obvious outliers may produce different credible algorithms, thereby affecting the selection of non-obvious outliers. Future research could explore ways of addressing these limitations and enhancing the proposed framework.

## REFERENCES


Alpaydin, E. 2020. *Introduction to machine learning*. Cambridge, MA: MIT Press.

Bao, Y., B. Ke, B. Li, Y. J. Yu, and J. Zhang. 2020. Detecting accounting fraud in publicly traded US firms using a machine learning approach. *Journal of Accounting Research* 58 (1): 199-235.

Breunig, M. M., H. P. Kriegel, R. T. Ng, and J. Sander. 2000. LOF: identifying density-based local outliers. *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data*: 93-104.

Craswell, N. 2009. Precision at N. *Encyclopedia of Database Systems*: 2127–2128. Available at: https://doi.org/10.1007/978-0-387-39940-9_484.

Denning, D. E. 1987. An intrusion-detection model. *IEEE observations on software engineering* (2):222-232.

Hawkins, D. M. 1980. *Identification of outliers*. Vol. 11: Springer.

Hevner, A.R., S. T. March, J. Park, and S. Ram. 2004. Design Science in Information Systems Research. *MIS Quarterly 28*, no. 1: 75–105. Available at: https://doi.org/10.2307/25148625.

Hodge, V., and J. Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22: 85-126.

Jabez, J., and B. Muthukumar. 2015. Intrusion detection system (IDS): anomaly detection using outlier detection approach. *Procedia Computer Science* 48: 338-346.

Malini, N., and M. Pushpa. 2017. Analysis on credit card fraud identification techniques based on KNN and outlier detection. Paper read at *3rd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics*. Chennai, India: AEEEICB.

No, W. G., K. Lee, F. Huang, and Q. Li. 2019. Multidimensional audit data selection (MADS): A framework for using data analytics in the audit data selection process. *Accounting Horizons* 33(3): 127-140.

Pillai, S. K., M. M. Raghuwanshi, and M. Gaikwad. 2020. Hyperparameter tuning and optimization in machine learning for species identification system. In *Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019), NITTTR Chandigarh, India*, 235-241. Springer Singapore.

Ramaswamy, S., R. Rastogi, and K. Shim. 2000. Efficient algorithms for mining outliers from large data sets. Paper read at *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Dallas, Tx: SIGMOD.

Ramasamy, L. K., S. Kadry, and S. Lim. 2021. Selection of optimal hyper-parameter values of support vector machine for sentiment analysis tasks using nature-inspired optimization methods. *Bulletin of Electrical Engineering and Informatics* 10(1): 290-298.

Sejr, J. H., and A. Schneider-Kamp. 2021. Explainable outlier detection: What, for Whom and Why? *Machine Learning with Applications* 6: 100-172.

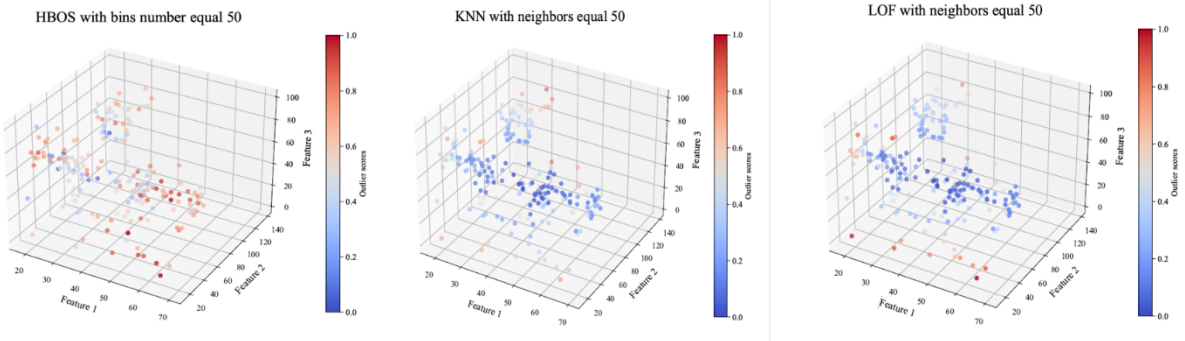
Simon, H. A. 1996. The Science of the Artificial (3rd ed.). Cambridge, Mass.: MIT Press.

Thiprungsri, S., and M. A. Vasarhelyi. 2011. Cluster Analysis for Anomaly Detection in Accounting Data: An Audit Approach. *International Journal of Digital Accounting Research* 11.

van Capelleveen, G. V., M. Poel, R. M. Mueller, D. Thornton, and J. van Hillegersberg. 2016. Outlier detection in healthcare fraud: A case study in the Medicaid dental domain. *International Journal of Accounting Information Systems* 21: 18-31.

Zimek, A., R. J. Campello, and J. Sander. 2014. Ensembles for unsupervised outlier detection: challenges and research questions [Position Paper] Arthur. *Acm Sigkdd Explorations Newsletter* 15(1): 11-22.

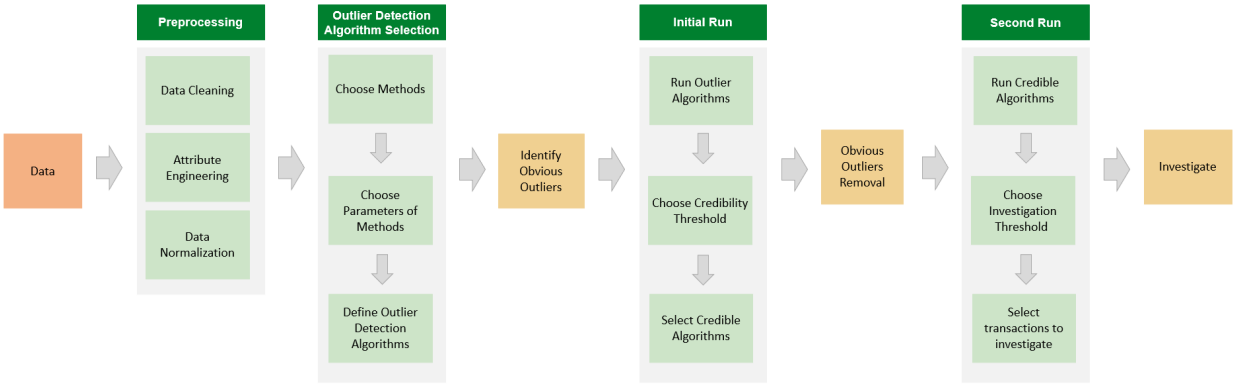
Figure 1. Outlier scores calculated by HBOS, KNN and LOF based on Mall-Customer-Segmentation dataset[2][3]




[2] The Mall-Customer-Segmentation dataset (https://github.com/jeffrey125/Mall-Customer-Segmentation) contains information about customer profiles entering a Mall: customer age (Attribute 1), customer's annual income (Attribute 2), and customer's expenditure capability (Attribute 3). In this application, outlier detection methods can identify anomalous customer behaviors inconsistent with the regular ones.

[3] The color bar on the right side indicates that a dark red point denotes a greater outlier score (higher likelihood of being an outlier), whereas blue points identify normal observations. Different methods often generate different results identifying outliers.

Figure 2. Outlier detection audit application framework



# APPENDIX

## Appendix A

AU-C Section 520 (AICPA) defines analytical procedures as "*Evaluations of financial information through analysis of plausible relationships among both financial and nonfinancial data. Analytical procedures also encompass such investigation, as is necessary, of identified fluctuations or relationships that are inconsistent with other relevant information or that differ from expected values by a significant amount."*[4]

AS 2305 (PCAOB) defines analytical procedures as *"... an important part of the audit process and consist of evaluations of financial information made by a study of plausible relationships among both financial and nonfinancial data. Analytical procedures range from simple comparisons to the use of complex models involving many relationships and elements of data. A basic premise underlying the application of analytical procedures is that plausible relationships among data may reasonably be expected to exist and continue in the absence of known conditions to the contrary."*[5]

[4] https://us.aicpa.org/content/dam/aicpa/research/standards/auditattest/downloadabledocuments/au-c-00520.pdf
[5] https://pcaobus.org/oversight/standards/auditing-standards/details/AS2305

## Appendix B

**Table 1. 23 Detailed outlier detection survey articles between 2003 and 2022[6]**

| Cites | Authors | Title | Year |
|---|---|---|---|
| 12589 | V Chandola, A Banerjee, V Kumar | Anomaly detection: A survey | 2009 |
| 4416 | V Hodge, J Austin | A survey of outlier detection methodologies | 2004 |
| 2042 | A Patcha, JM Park | An overview of anomaly detection techniques: Existing solutions and latest technological trends | 2007 |
| 1938 | M Markou, S Singh | Novelty detection: a review—part 1: statistical approaches | 2003 |
| 1652 | MAF Pimentel, DA Clifton, L Clifton, L Tarassenko | A review of novelty detection | 2014 |
| 1419 | MH Bhuyan, DK Bhattacharyya… | Network anomaly detection: methods, systems and tools | 2013 |
| 1393 | L Akoglu, H Tong, D Koutra | Graph based anomaly detection and description: a survey | 2015 |
| 1319 | EWT Ngai, Y Hu, YH Wong, Y Chen, X Sun | The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature | 2011 |
| 1268 | M Ahmed, AN Mahmood, J Hu | A survey of network anomaly detection techniques | 2016 |
| 1124 | M Markou, S Singh | Novelty detection: a review—part 2: neural network based approaches | 2003 |
| 1103 | C Phua, V Lee, K Smith, R Gayler | A comprehensive survey of data mining-based fraud detection research | 2010 |
| 1054 | G Pang, C Shen, L Cao, AVD Hengel | Deep learning for anomaly detection: A review | 2021 |

[6] The software tool Publish or Perish (https://harzing.com/resources/publish-or-perish) gathers and analyzes academic citations. It employs a variety of data sources to collect the raw citations, analyzes them, and then offers a wide range of citation metrics, such as the number of papers, total citations, and the h-index.

| | | | |
|---|---|---|---|
| 1042 | M Gupta, J Gao, CC Aggarwal | Outlier detection for temporal data: A survey | 2013 |
| 999 | Y Zhang, N Meratnia, P Havinga | Outlier detection techniques for wireless sensor networks: A survey | 2010 |
| 882 | A Zimek, E Schubert, HP Kriegel | A survey on unsupervised outlier detection in high-dimensional numerical data | 2012 |
| 712 | V Chandola, A Banerjee, V Kumar | Anomaly detection for discrete sequences: A survey | 2010 |
| 572 | Y Zhao, Z Nasrullah, Z Li | Pyod: A python toolbox for scalable outlier detection | 2019 |
| 463 | L Ruff et al. | A unifying review of deep and shallow anomaly detection | 2021 |
| 447 | S Seo | A review and comparison of methods for detecting outliers in univariate data sets | 2006 |
| 433 | R Domingues, M Filippone, P Michiardi, J Zouaoui | A comparative evaluation of outlier detection algorithms: Experiments and analyses | 2018 |
| 396 | V Chandola, A Banerjee, V Kumar | Outlier detection: A survey | 2007 |
| 325 | H Wang, MJ Bah, M Hammad | Progress in outlier detection techniques: A survey | 2019 |
| 169 | A Boukerche, L Zheng, O Alfandi | Outlier detection: Methods, models, and classification | 2020 |

**Table 2. Brief descriptions of popular outlier detection methods[7]**

| **Categories** | **Representative Methods** | | |
|---|---|---|---|
| | **Method name** | **Literature** | **Rationale** |

[7] We do not provide detailed reviews of the methods but discuss instead the rationales of these methods and present an overall picture of popular outlier detection approaches.

| | | | |
|---|---|---|---|
| Statistics-based | Angle-based outlier detection (ABOD) | Kriegel et al. (2008) | ABOD presumes outliers are distant from the remaining points and have a small range of angles between pairs of points, whereas normal objects have a greater range. |
| | Histogram-based Outlier Score (HBOS) | Goldstein and Dengel (2012) | For HBOS, if numerous attribute values of a data point are out of the ordinary, it is likely an outlier. |
| | Principal component analysis (PCA) | Shyu, Chen, Sarinnapakorn, and Chang. (2003) | PCA reduces data dimensionality and the resultant principal components capture as much information as possible about regular points. Hence, a data point tends to be an outlier if there is a considerable gap between the original item and the one reconstructed using principal components. |
| | Minimum Covariance Determinant (MCD) | Hardin and Rocke 2004 | MCD finds a subset of points that are the least outlying. Then, a point is considered an outlier if its Mahalanobis distance to the least outlying points exceeds a specific threshold. |
| | Gaussian mixture model (GMM) | Zhuang et al. (1996) | By assuming points within a data set are generated from multiple Gaussian distributions, GMM identifies data points with low probability of belonging to dominant distributions as outliers. |

| | | | |
|---|---|---|---|
| Distance-based | K-nearest neighbor (KNN) | Ramaswamy et al. (2000) | For KNN, if a data point is geometrically farther away from its neighboring points than other points, it is likely to be an outlier. |
| Density-based | Local outlier factor (LOF) | Breunig et al. (2000) | LOF identifies as outliers the data points with significantly lower local density than their neighbors. |
| Clustering-based | Clustering-based Local Outlier Factor (CBLOF) | He, Xu, and Deng. (2003) | CBLOF initially clusters data points based on their feature similarities. Then, the points (1) located in small clusters that are geographically isolated, or (2) positioned in large clusters but far from centroids are considered outliers. |
| Deep learning-based | Autoencoder (AE) | Aggarwal (2017) | AE contains three categories of successively connected layers: the input layer, the hidden layers, and the output layer. This technique reconstructs the input from the output as closely to the original input as possible. The data points with significant reconstruction errors are considered to be outliers. |
| Ensemble-based | Isolation forest (IF) | Liu, Ting, and Zhou. (2008) | IF performs random splits on attribute values based on a single tree model to separate anomalous data points from normal ones. Eventually, anomalous predictions from multiple tree models are integrated to produce robust mining results. |

**Table 3. Parameter configurations utilized by outlier detection methods[8]**

| Method | Hyperparamer 1 | Hyperparamer 2 | Hyperparamer 3 | Total |
|---|---|---|---|---|
| HBOS | n_histograms: [5, 10, 20, 30, 40, 50, 75, 100] | tolerance: [0.1, 0.2, 0.3, 0.4, 0.5] | N/A | 40 |
| PCA | n_components: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] | N/A | N/A | 18 |
| MCD | support_fraction: [0.5, 0.55 ,0.6, 0.65 ,0.7, 0.75 ,0.8, 0.85, 0.9 ,0.95 ,1] | N/A | N/A | 11 |
| KNN | n_neighbors: [1, 5, 10, 15, 20, 25, 50, 60, 70, 80, 90, 100] | method: [largest, mean, median] | N/A | 36 |
| LOF | n_neighbors: [1, 5, 10, 15, 20, 25, 50, 60, 70, 80, 90, 100] | distance: [Manhattan, euclidean, minkowski] | N/A | 36 |
| CBLOF | n_clusters: [5, 8, 15, 20] | alpha: [0.8, 0.9, 0.95] | beta: [2, 4, 5] | 34 |
| AE | hidden_neurons: [64, 36, 18, 8, 18, 36, 64], [36, 18, 9, 18, 36], [24, 12, 6, 12, 24] | Epochs: [50, 100, 200] | N/A | 9 |
| IF | n_estimators: [10, 20, 30, 40, 50, 75, 100, 150, 200] | max_features: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] | N/A | 81 |

Note: we completed the evaluation case study using version 1.0.9 of the Python PyOD package. Please check the package official document[9] for the definitions of relevant hyperparameters.

[8] In evaluating the proposed outlier detection framework on the revenue dataset, the results of the initial run show that, with a loose credibility threshold fixed at nine, 32 outlier detection algorithms are selected (18 PCA algorithms with the number of components ranging from one to eighteen, five MCD algorithms with the support_fraction equal 0.55, 0.6, 0.65, 0.85, and 0.9, as well as nine Autoencoder algorithms). With a stringent credibility threshold fixed at 18, only three outlier detection algorithms passed the credibility test (three MCD methods with the support_fraction equal 0.55, 0.6, and 0.65).

[9] https://pyod.readthedocs.io/en/latest/

Notably, aside from the essential hyperparameters listed in Table 3, all other parameters used by each method are set to their default values.

**Table 4. Data considerations in the algorithm selection process**

| | |
|---|---|
| 1 | Algorithm performance varies with data sizes. Some algorithms are robust on large datasets, while others can be slow during execution. Auditors should select algorithms that are suitable for the size of data to be analyzed. |
| 2 | High dimensionality can lead to high complexity, which can be problematic for certain methods. Therefore, auditors should consider the data dimensionality in selecting algorithms. |
| 3 | Data distribution is usually not an issue for outlier detection methods. However, ABOD will not function on datasets containing duplicate observations. |

## Appendix C. Preprocessing

Observation data often needs to be converted to suitable formats in order to apply outlier detection algorithms. The conversion includes data cleaning, attribute engineering, data normalization, etc. (García, Luengo, and Herrera 2015). The three preprocessing steps are summarized in this part.

### *Cleaning*

Raw data may contain irregularities such as missing values, which many outlier detection methods cannot handle. Accordingly, observations that contain missing values can be dropped, or the values filled in (García et al. 2015); attributes that contain multiple missing values can be ignored. Removal of certain attributes may artificially cause the appearance of duplicate observations, which some outlier detection methods (e.g., ABOD) cannot handle. An approach of dealing with duplicates by retaining a single representative of every subset of duplicate observations may not

be appropriate for outlier detection, because the prevalence of duplicates may be indicative of these observations being normal.

### *Attribute Engineering*

Irrelevant, non-essential attributes or attributes that contain many missing values can be filtered out, leading to simpler, robust models (Alpaydin 2020). Auditors may construct new attributes to capture more focused information for a specific dataset (e.g., calculating the difference between the observation and bookkeeping dates can simplify the identification of backdating). Specific objectives should guide the auditors in engineering new attributes.

### *Normalization*

Normalization can improve outlier detection performance (Campos et al. 2016). The aim is that all variables have the same spread and are comparable to each other.

## Appendix D. Unsupervised vs. Supervised Learning in ADA

| **Supervised Learning** | |
|---|---|
| **Data** | A model is trained on a labeled dataset, e.g., labeled transactions indicating anomalies. |
| **Goal** | To label new transactions. |
| **Examples in ADA**: | Classification (dividing risky and non-risky transactions) and regression (predicting a risk score). |
| **Unsupervised Learning** | |
| **Data** | A model is run on an unlabeled dataset, e.g., unlabeled transactions. |
| **Goal** | To discover patterns, relationships, or structures within the data. No "correct" label to guide the learning process. |
| **Examples in ADA:** | Clustering (grouping similar transactions) and outlier detection (detecting abnormal transactions). |

Different from supervised learning, unsupervised learning does not utilize labeled data for training and validation. Instead, a model is constructed solely by applying a mathematical

algorithm to unlabeled data. As the data is not labeled, there is no way to execute a separate validation step to assess the performance of the model. Therefore, cross-validation, which is predominantly used in supervised learning to assess how well a model will generalize to new data, cannot be applied in unsupervised learning.

## Appendix E. "Weakly" Supervised Algorithm Selection

We introduce the term "weakly" supervised to denote an approach based on unsupervised learning that utilizes an extremely small number of labeled observations for algorithm selection. This term is similar to the concept of incomplete supervision in the extant literature (Zhou 2018). The design employs auditors' judgment on "obvious outliers" within transaction data. In practice, only very few transactions are labeled in this way. Though predominantly unsupervised, the incorporation of "obvious outliers" warrants the "weakly" supervised label for the proposed framework. Compared with the goals set in Appendix D, "weakly" supervised learning refers to the design wherein algorithms discover patterns, relationships, or structures within the data by leveraging a small set of labeled transactions.

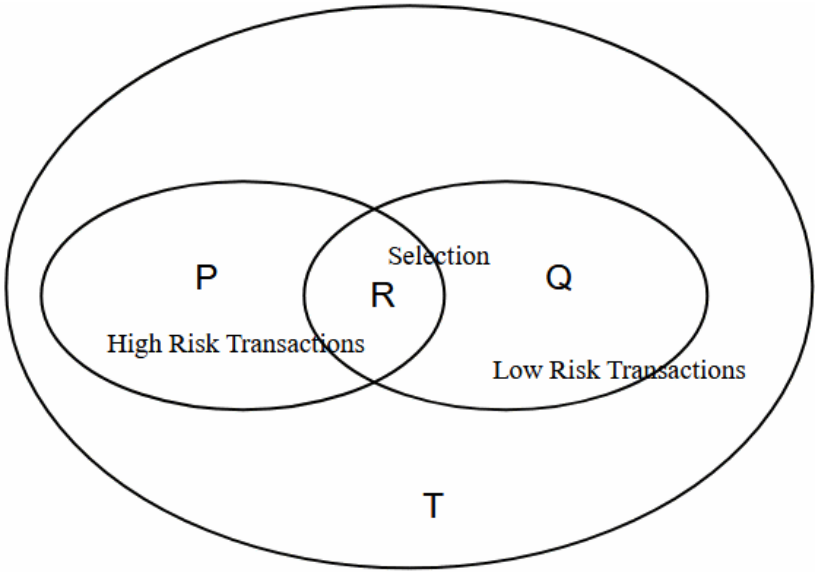
## Appendix F. Precision at n

Precision at n (P@n) is an evaluation metric used when only a limited number of observations can be examined. Specifically, n is chosen based on an assumption about the number of transactions that auditors will examine, and the number of risky transactions (i.e., true positives) among the top-n transactions is a measure of selection success. P@n is calculated as the percentage of risky transactions captured by an algorithm in its top-n transactions selected based on their outlier scores.

A given dataset of N transactions can be viewed as a disjoint union of L low risk transactions (true inliers) and H high risk transactions (true outliers), so that N = L + H. An outlier detection algorithm selects n transactions based on their outlier scores. As shown in Figure 3, this selection is a disjoint union of R true outliers (true positives) and Q true inliers (false positives), so that n = R + Q. The set of transactions that are not selected is also a disjoint union of P true outliers (false negatives) and T true inliers (true negatives). Note that P + R = H and Q + T = L. After the examination of the selected n transactions by the auditors, the counts R and Q will be determined. The counts P and T will remain unknown since those transactions that are not selected will not be examined. A selection is superior if it results in fewer false positives (Q) and fewer false negatives (P).

It often happens in machine learning that decreasing false positives is associated with increasing false negatives, and vice versa. However, in the application of outlier detection algorithms discussed in this paper, the selection size n has to be the same for every algorithm. Since n = R + Q, minimizing false positives (Q) is equivalent to maximizing true positives (R). Since P + R = H, and H depends only on the given dataset, and therefore it is the same no matter what outlier detection algorithm is used, maximizing true positives (R) is equivalent to minimizing false negatives (P). Therefore, maximizing true positives results in minimizing both false positives and false negatives. Consequently, the algorithm with the highest P@n performs the best in identifying risky transactions in a given dataset.

***Figure 3. Venn diagram for P@n***




## Appendix G. Cross-validation

Cross-validation is a resampling procedure used to evaluate the accuracy of supervised machine learning models when the amount of available labeled data is limited. A commonly used type of of cross-validation, k-fold cross-validation involves dividing a dataset into 'k' subsets (Alpaydin 2020). The model is trained on 'k-1' of these folds and tested on the remaining one. This process is repeated 'k' times with a different fold used for testing each time, and the average over the k experiments is used as the estimate of accuracy.

To employ supervised learning and cross-validation, the auditors have to classify each transaction as anomalous or not. However, in real-world scenarios, it is challenging for auditors to label a sufficiently large sample to construct a well-trained model. Therefore, supervised learning, and consequently, cross-validation is not commonly used in audit practice.

## Appendix H. Datasets Utilized in the Paper

**Table 5. Datasets utilized in this paper**

| Dataset Name | Dataset Size | Number of attributes after preprocessing and attribute engineering | Number of obvious outliers | Credibility Threshold | | Number of Credible Algorithms[10] | |
|---|---|---|---|---|---|---|---|
| | | | | Loose | Stringent | Loose | Stringent |
| Archival Procurement dataset | 56,487 | 23 | 35 | - | - | - | - |
| Archival revenue dataset | 344,593 | 18 | 48 | 9 | 18 | 32 | 3 |
| Live engagement revenue dataset | 36,490 | 14 | 32 | 9 | 12 | 33 | 10 |

- The procurement dataset has 56,487 observations and contains a wide range of observation attributes, like vendor profile information, payment amount, and observation occurrence time.
- The archival revenue dataset consists of 344,593 observations described by a broad spectrum of observation characteristics (e.g., date, dollar value, type, and customer profile information). Data preprocessing procedures are required to transform raw observation data into appropriate formats prior to applying outlier detection methods. Using engagement-specific knowledge obtained from a practicing auditor, some non-essential attributes are removed, and the attribute engineering technique is utilized to create 18 attributes that capture the targeted dataset information. The final attribute set includes observation-type dummy variables, indicators of unusual vendor addresses, weekdays, etc.
- The live engagement revenue dataset has 36,490 observations and similar attributes as the archival revenue dataset, so we applied similar preprocessing and attribute engineering procedures to it and created 14 attributes used in subsequent processing.

[10] The loose selection included Autoencoder, PCA, and MCD algorithms, while the stringent selection included MCD algorithms only.

## References in the Appendix[11]


Aggarwal, Charu C. 2017. *An introduction to outlier analysis*. Springer International Publishing.

Ahmed, M., A. N. Mahmood, and J. Hu. 2016. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications* 60: 19-31.

Akoglu, L., H. Tong, and D. Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery* 29: 626-688.

Alpaydin, E. 2020. *Introduction to machine learning*. Cambridge, MA: MIT Press.

Bao, Y., B. Ke, B. Li, Y. J. Yu, and J. Zhang. 2020. Detecting accounting fraud in publicly traded US firms using a machine learning approach. *Journal of Accounting Research* 58 (1): 199-235.

Bhuyan, M. H., D. K. Bhattacharyya, and J. K. Kalita. 2013. Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials* 16 (1): 303-336.

Boukerche, A., L. Zheng, and O. Alfandi. 2020. Outlier detection: Methods, models, and classification. *ACM Computing Surveys (CSUR)* 53 (3): 1-37.

Breunig, M. M., H. P. Kriegel, R. T. Ng, and J. Sander. 2000. LOF: identifying density-based local outliers. *In Proceedings of the 2000 ACM SIGMOD international conference on Management of data*: 93-104.

Campos, G.O., A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery* 30: 891-927.

Chandola, V., A. Banerjee, and V. Kumar. 2007. Outlier detection: A survey. *ACM Computing Surveys* 14: 15.

Chandola, V., A. Banerjee, and V. Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41 (3): 1-58.

Chandola, V., A. Banerjee, and V. Kumar. 2010. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering* 24 (5): 823-839.

Domingues, R., M. Filippone, P. Michiardi, and J. Zouaoui. 2018. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition* 74: 406-421.

García, S., J. Luengo, and F. Herrera. 2015. *Data preprocessing in data mining* (Vol. 72, 59-139). Cham, Switzerland: Springer Press.

Goldstein, M. and A. Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, *1*, 59-63.

Gupta, M., J. Gao, C. C. Aggarwal, and J. Han. 2013. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering* 26 (9): 2250-2267.

Hardin, J. and D. M. Rocke. 2004. Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator. *Computational Statistics & Data Analysis*, *44*(4), 625-638.

He, Z., X. Xu, and S. Deng. 2003. Discovering cluster-based local outliers. *Pattern recognition letters*, *24*(9-10), 1641-1650.

Hodge, V., and J. Austin. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22: 85-126.


[11] The references are based on the results in Google Scholar. The title and authorship sequence can be slightly different from Publish or Perish result in the table.

Kriegel, H. P., M. Schubert, and A. Zimek. 2008. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 444-452).

Liu, F. T., K. M. Ting, and Z. H. Zhou.2008. Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413-422). IEEE.

Markou, M., and S. Singh. 2003. Novelty detection: a review—part 1: statistical approaches. *Signal Processing* 83 (12): 2481-2497.

Markou, M., and S. Singh. 2003. Novelty detection: a review—part 2: neural network based approaches. *Signal Processing* 83 (12): 2499-2521.

Ngai, E. W. T., Y. Hu, Y. H. Wong, Y. Chen, and X. Sun. 2011. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems* 50 (3): 559-569.

Pang, G., C. Shen, L. Cao, and A. Van Den Hengel. 2021. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)* 54 (2): 1-38.

Patcha, A., and J-M. Park. 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks* 51 (12): 3448-3470.

Phua, C., V. Lee, K. Smith, and R. Gayler. 2010. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*.

Pimentel, M. A. F., D. A. Clifton, L. Clifton, and L. Tarassenko. 2014. A review of novelty detection. *Signal Processing* 99: 215-249.

Ramaswamy, S., R. Rastogi, and K. Shim. 2000. Efficient algorithms for mining outliers from large data sets. Paper read at *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Dallas, Tx: SIGMOD.

Ruff, L., J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K-R. Müller. 2021. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE* 109 (5): 756-795.

Seo, S. 2006. A review and comparison of methods for detecting outliers in univariate data sets. PhD diss., University of Pittsburgh.

Shyu, M. L., S. C. Chen, K. Sarinnapakorn, and L. Chang. 2003. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop* (pp. 172-179). IEEE Press.

Wang, H., M. J. Bah, and M. Hammad. 2019. Progress in outlier detection techniques: A survey. *IEEE Access* 7: 107964-108000.

Zhang, Y., N. Meratnia, and P. Havinga. 2010. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials* 12 (2): 159-170.

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

Zhou, Zhi-Hua. 2018 A brief introduction to weakly supervised learning. *National science review* 5 (1): 44-53.

Zhuang, X., Y. Huang, K. Palaniappan, and Y. Zhao. 1996. Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing* 5 (9): 1293-1302.

Zimek, A., E. Schubert, and H-P. Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5 (5): 363-387.