technologies

MDPI

Review

Cleaning Big Data Streams: A Systematic Literature Review

Obaid Alotaibi [1,2], Eric Pardede [2,*] and Sarath Tomy [3]

[1] Department of Computer Science, College of Science and Arts, Sajir Campus, Shaqra University, Sajir City 11951, Saudi Arabia; obaid@su.edu.sa
[2] Department of Computer Science and Information Technology, School of Engineering and Mathematical Sciences, Melbourne Campus, La Trobe University, Melbourne, VIC 3086, Australia
[3] Department of Computer Science and Information Technology, School of Engineering and Mathematical Sciences, Bendigo Campus, La Trobe University, Bendigo, VIC 3552, Australia; s.tomy@latrobe.edu.au
* Correspondence: e.pardede@latrobe.edu.au

Abstract: In today's big data era, cleaning big data streams has become a challenging task because of the different formats of big data and the massive amount of big data which is being generated. Many studies have proposed different techniques to overcome these challenges, such as cleaning big data in real time. This systematic literature review presents recently developed techniques that have been used for the cleaning process and for each data cleaning issue. Following the PRISMA framework, four databases are searched, namely IEEE Xplore, ACM Library, Scopus, and Science Direct, to select relevant studies. After selecting the relevant studies, we identify the techniques that have been utilized to clean big data streams and the evaluation methods that have been used to examine their efficiency. Also, we define the cleaning issues that may appear during the cleaning process, namely missing values, duplicated data, outliers, and irrelevant data. Based on our study, the future directions of cleaning big data streams are identified.

Keywords: clean; big data; stream; machine learning; deep learning; artificial intelligence; missing value; outliers; duplicate data; irrelevant data

Citation: Alotaibi, O.; Pardede, E.; Tomy, S. Cleaning Big Data Streams: A Systematic Literature Review. *Technologies* **2023**, *11*, 101. https://doi.org/10.3390/technologies11040101

Academic Editors: Sikha Bagui and Mauro Iacono

Received: 22 May 2023
Revised: 21 July 2023
Accepted: 24 July 2023
Published: 26 July 2023

1. Introduction

In today's big data era, streaming data has recently received increased attention from researchers, especially in relation to cleaning and analyzing streaming data. Streaming data refer to data which are continuously generated from various sources and moving them to a particular destination. The accumulation of these sources generates what is called a big data stream.

Big data is often described by five characteristics, namely volume, velocity, variety, veracity, and value Erl et al. [1], which differentiate it from traditional data. Volume refers to the size, scale, quantity, and magnitude of the data that have been generated, the size of which can exceed hundreds of terabytes. Velocity refers to the arrival speed of data, which results in the accumulation of enormous datasets within very short periods. Variety refers to the different formats of data, namely structured data, semi-structured data, and unstructured data. Structured data have a standardized format and a well-defined structure and generally reside in relational databases to represent the relationship between entities such as tables. Semi-structured data are not as rigid as structured data; however, they have several elements that are similar, being organized hierarchically, although it cannot be verified whether the tabular structure is associated with relational databases. Unstructured data, which are growing at a faster rate than structured and semi-structured data, have no easily identifiable structure and they cannot be stored in any logical form Erl et al. [1]. Veracity refers to the validity or quality of data. Value refers to the usefulness of data for decision making. Big data's characteristics increase the difficulty of the cleaning process compared with that for normal data due to the heterogeneous data, the volume, and

speed of arrival data, which reveal a limitation of using the normal data cleaning method. However, the normal data cleaning methods can be considered as a baseline for developing cleaning methods able to cope with big data's characteristics.

Decision making is the ability to make decisions based on data analysis. Data analytics is the task of extracting useful information from data after they have been processed. There are four types of data analytics, descriptive, diagnostic, predictive, and prescriptive Erl, et al. [1], which use different algorithms and techniques for analysis. Descriptive analytics provides an insight into what happened based on a summary of existing data. Diagnostic analytics provides a deep understanding of a specific situation and finds reasons for why something happened. Predictive analytics is used to analyze past data patterns and trends to predict what could happen in the future. Prescriptive analytics is undertaken to determine which decision option, given multiple options, is the best in terms of taking advantage of future opportunities or avoiding risks, based on the data available. Data analysis involves data processing, of which there are two types, stream processing and batch processing. Batch processing involves processing data after they have been collected, whereas stream processing involves the real-time processing of continuous streams of data in motion, which are then passed off to a specific destination. A brief comparison of these two types of data processing is shown in Table 1.

Table 1. A comparison of types of processing data.

Aspect	Stream Processing	Batch Processing
Data size	Unknown	Known
Performance	Limited time, it can be seconds or milliseconds	No limit, it can be hours or days
Dataset type	Unbounded	Bounded
Processing	It is processed only once	It can be processed many times
Example	E-commerce transactions	Payroll system

There are more difficulties associated with data stream processing than batch processing. In addition, big data streaming analytics faces many challenges due to the characteristics of big data. For instance, it is difficult to apply existing techniques and data mining tools to big data stream analysis because of the speed at which data arrive Kolajo et al. [2] and also the variety of data formats. Understanding how to deal with these and extract useful information from the data in real-time Kolajo et al. [2] is a further challenge. Figure 1 visualizes the streaming data lifecycle.

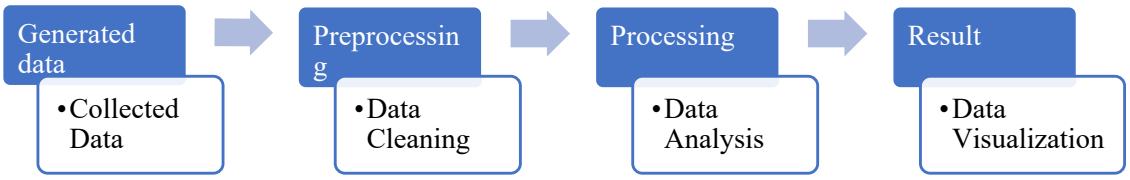

Figure 1. Streaming data lifecycle.

It is necessary to process big data streams in real time so organizations and businesses can react to changes in conditions in real time Kolajo et al. [2]. For example, by undertaking the real-time analysis of big data generated by the Internet of Things (IoT) devices in areas such as smart cities, oil and gas, transportation, and manufacturing can help to detect anomalous data and identify problematic issues in the early stage; hence, the problem can be fixed before it becomes a bigger and more costly issue to address. There is a direct correlation between data analysis and data cleaning, so more clean data will lead to an increase in the quality of data and will result in more trustworthy and accurate results that will help businesses to make better decisions based on data analysis. Hence, data cleaning plays a major role in increasing data quality.

Data cleaning tackles many problems in data that have been generated by humans or machines, such as missing values, duplicated data, outliers, or irrelevant data. Missing data are defined as a data value which is not stored for a variable in the observation under investigation. There are several methods for dealing with missing data, such as ignoring them, imputation, and deleting the missing value. Ignoring missing data means no action is taken. Imputation means a value is assigned to the missing data, for example, the median if the missing data type is numerical or "Unknown" if the missing data type is categorical. Deletion means removing the entire row of data that contains a missing value. Data duplication is where the same data are repeated more than once in the dataset, which leads to an increase in the size of the dataset. Data cleaning uses a deduplication method to tackle duplicated data by keeping one row of the duplicated data and removing the rest of the repeated data.

An outlier is unnormal data also known as anomalous data. There are three types of outliers, namely global outliers, contextual outliers, and collective outliers Han et al. [3]. Global outliers can be defined as a data point that deviates significantly from the overall distribution of the dataset. A contextual outlier is a data point whose value significantly deviates from the other data points in the same context Han et al. [3]. For example, if a dataset shows that the temperature in Melbourne is 32 °C, this could be a contextual outlier if the season in Melbourne is winter; however, if the season is summer, then this can be considered normal data since the temperature in Melbourne can reach 32 °C or higher in summer. Collective outliers are groups of data points that deviate significantly from the rest of the data. It is important to note that individual data objects may not be outliers, but the objects as a whole deviate significantly from the entire dataset Han et al. [3]. For instance, a delay in the delivery of 160 items on a single day is considered to be a collective outlier; however, a delay in the delivery of a single item on a particular day is normal Han et al. [3]. Outliers can be handled in two ways, either by removing them or keeping them. Irrelevant data are unwanted data and should not be analyzed in certain situations; thus, they should be excluded before data analysis commences. Figure 2 illustrates a simple example for each issue that may appear in the data cleaning process in a structured data format, and Figure 3 depicts the possible solutions for data cleaning.

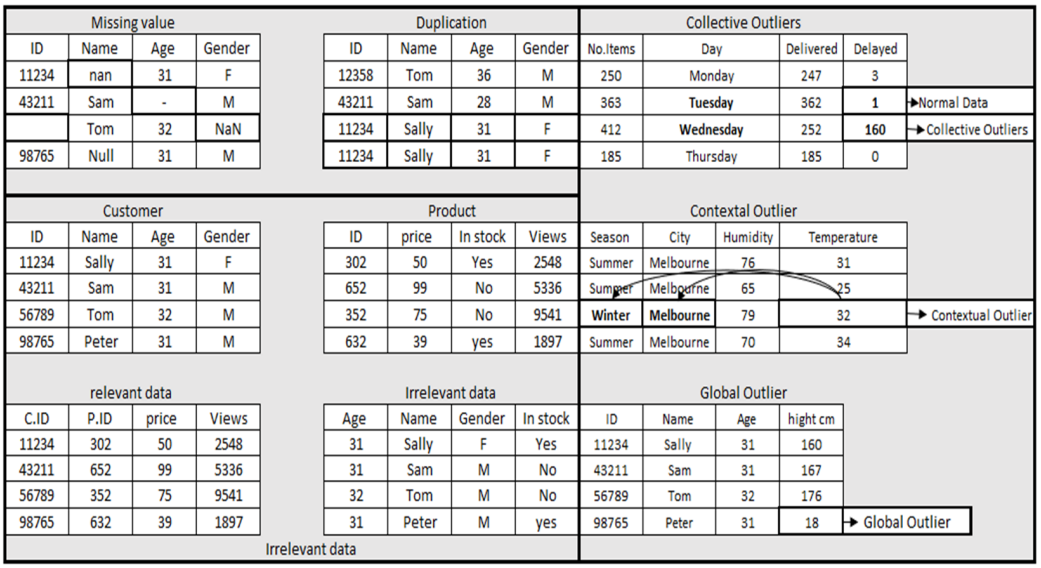

Figure 2. Data cleaning issues.

The issues which are encountered during the data cleaning process can affect data of all formats. Therefore, data cleaning to fix or remove incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset is essential prior to data analysis to ensure organizations can engage in quality decision making.

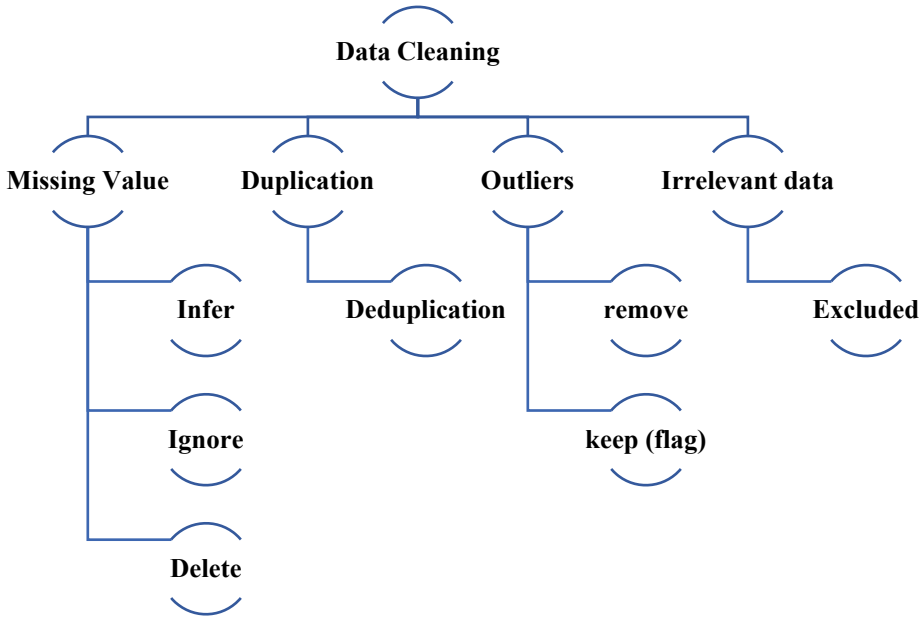

Figure 3. Possible solutions for data cleaning.

The increased use of the Internet and technology has resulted in a generation of massive data streams which need to be cleaned in real time or near real time. In addition, many researchers have concentrated on analyzing big data streams rather than cleaning big data streams in real time Kolajo et al. [2]. Thus, more attention needs to be devoted to cleaning big data streams in real time because the accuracy of data analysis depends on the quality of data cleaning. For instance, cleaning streaming data from the banking sector will increase the accuracy of credit card fraud detection and aid in its prevention. Unclean data gathered from different sources will affect the accuracy of data analytics Ridzuan et al. [4]. Although it is important that data cleaning is undertaken in real time, it presents many challenges due to the enormous volume of the data generated having a short arrival time and various data formats. In short, the characteristics of big streaming data have increased the complexity of data cleaning, particularly for semi-structured and unstructured data due to the nature of the data. Therefore, this systematic literature review (SLR) aims to provide insight into recently developed techniques that have been used for data cleaning and determine the future research directions in data cleaning.

The rest of this paper is organized as follows: Section 2 details the SLR questions and the SLR methodology; Section 3 presents the literature review on data cleaning; Section 4 analyzes the SLR strategy; and Section 5 summarizes and concludes the SLR.

2. Systematic Literature Review

This section consists of two subsections, the SLR research questions and the SLR search strategy. The research questions are formulated to identify the research that has been conducted and what specific areas need further exploration, while the search strategy is the method that is followed to find relevant studies on cleaning big data streams for inclusion in the SLR.

2.1. Systematic Literature Review Questions

- RQ1: Why is it important to clean data streams?
- RQ2: Which data cleaning issue is most commonly discussed during the data cleaning process?
- RQ3: What sort of techniques are commonly used to clean data?
- RQ4: What methods have been used to evaluate the proposed approaches?
- RQ5: What are the future directions for data stream cleaning?

2.2. Systematic Literature Review Search Strategy

An SLR is a methodology to find relevant studies in a specific area. We selected four databases, namely IEEE Xplore, ACM Library, Scopus, and ScienceDirect, to search for relevant papers. As we used four well-known databases to conduct this systematic review, we will cover the publishing venues because this is out of the review scope. We determined the terms to use in the search, namely big data, clean, stream, quality, outlier anomaly, abnormal, duplicate, redundant, irrelevant, ML (machine learning), DM (data mining), AI (artificial intelligence), missing data, missing value, and noise. We constructed five queries from these terms as follows:

1. "Big data" AND (Clean* OR Stream* OR quality);
2. "Big data stream*" AND (Clean* OR Outlier* OR anomal* OR abnormal* OR Duplicat* OR redund* OR Irrelevant);
3. "Big data stream*" AND (ML OR DM OR AI);
4. "Big data stream*" AND (Missing Value* OR Missing data);
5. "Big data stream*" AND Noise.

We used the wildcard * to cover as many studies as possible. For example, clean* means that the results contain the terms clean, cleaned, cleaning, or cleansing, and so on. However, some databases support "?" as a wildcard instead of "*", so we used different search methods which were appropriate for each database.

We ran the queries in the aforementioned databases. Table 2 shows the number of records that were retrieved from each database.

Table 2. Number of records for each query.

Query	IEEE Xplore	ACM Library	Scopus	ScienceDirect
1	12,609	17,275	233,243	57,495
2	420	68	1096	174
3	160	21	645	116
4	63	76	101	102
5	115	19	304	125
Total		324,227		

We limited the results from each query to the last five years from 2018 to 2023 to retrieve only the most recent work on cleaning big data streams due to the huge number of results retrieved from the first query. We further reduced the results by specifying that the terms in the first query should appear in the document title and abstract and index terms or author keywords. It should be noted that even though we limited the results to those obtained in the interval 2018 to 2023, the ACM Library in queries 1 and 4 and the IEEE Xplore in query 4 return results for the interval 2018–2022. Table 3 shows the results for each query after limiting the publication date to the last five years.

We used Covidence software because it has been designed for SLRs, especially for the PRISMA framework PRISMA [5], which is used in this SLR, as shown in Figure 4. The reason why we followed the PRISMA strategy instead of the snowballing strategy is because we aimed to present recently developed techniques in the data cleaning process, so the results of the snowballing strategy might not have suited the purpose. We uploaded the results (5013 studies) into Covidence. One of the advantages of Covidence is that it removes duplicated studies (1136 research) by default. We screened the titles and abstracts of each of the retrieved 3877 studies. We excluded surveys, chapters, and papers that were not related to data cleaning; hence, 3715 studies were excluded, which left 162 studies after the screening phase. The eligibility criteria were as follows: the study must be written in the English language; it must be a complete study with an introduction, discussion of previous work, details of the methodology used, results, and a conclusion; and it must have reliable references. Table 4 shows the eligibility criteria that we applied in this stage. As a result, of the 164 studies, 76 were eliminated, leaving 86 studies on cleaning data.

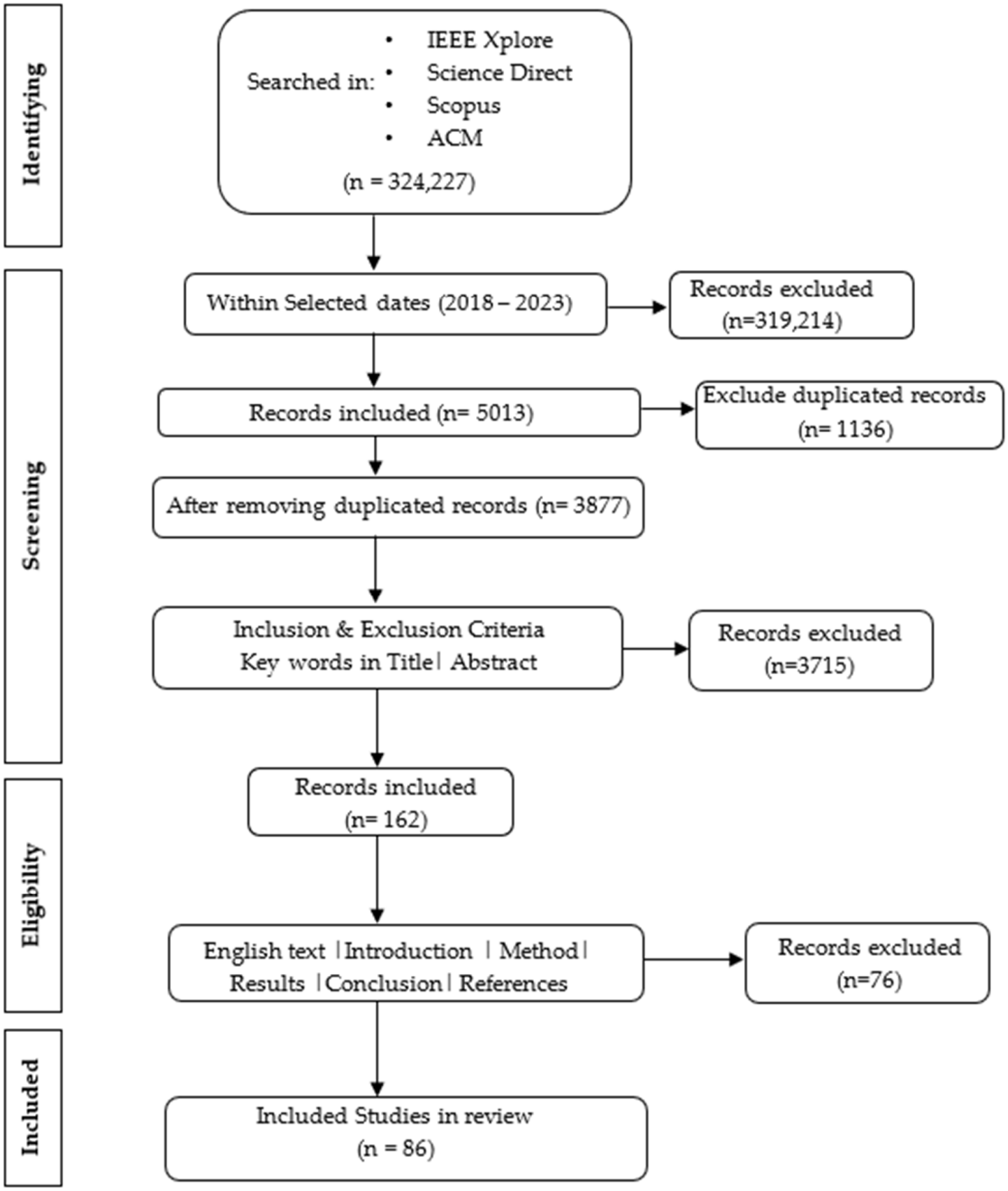

Figure 4. PRISMA framework.

Table 3. Number of records retrieved after limiting the publication date from 2018 to 2023.

Query	IEEE Xplore	ACM Library	Scopus	ScienceDirect
1	157	18	496	1505
2	271	34	920	137
3	142	13	568	99
4	38	37	90	77
5	68	12	241	90
Total		5013		

Table 4. Eligibility criteria.

Criteria	Eligible	Ineligible
Written language	English	Other languages
Study	Complete study	Incomplete study
References	Reliable references	Unreliable references

3. Literature Review

Many researchers have proposed several methods to clean data whether in streams or batches. Some of these approaches are an improvement or an extension of an existing method, a combination of methods, or a new approach aiming to address a specific data cleaning issue. These methods used diverse techniques, such as artificial intelligence, machine learning, deep learning, statistical methods, etc., to tackle issues that may appear during the data pre-processing stage, as discussed in the following subsections.

3.1. Artificial Intelligence

Several studies used artificial intelligence methods to deal with data cleaning problems. Turabieh et al. [6] suggested a dynamic-adaptive-network-based fuzzy inference system to deal with missing values. The hepatitis dataset and mammographic mass were used to examine the proposed technique and the results were compared to existing methods in terms of area under the curve and statistical measures. Sun et al. [7] used edge intelligence to propose a data stream cleaning system for global outlier detection. Data were collected from injection molding machines and through monitoring base stations. The results of the proposed method were compared to a similar existing system.

3.2. Machine Learning

Several studies have proposed machine learning approaches to address data cleaning issues. Shao et al. [8] proposed a technique based on shared nearest-neighbor density to detect global outliers. The Third International Knowledge Discovery and Data Mining Tools Competition (KDD-CUP-99) dataset was used in the experiment, and the result was compared to a D-Stream algorithm, where the proposed algorithm had better results than the D-Stream algorithm in terms of clustering quality and runtime. Vázquez et al. [9] proposed an algorithm called sparse data observers based on low-density models for global outlier detection. A total of 12 semantic datasets and 11 literature datasets were used to examine the efficiency of the proposed algorithm, and the results were compared to the local outlier factor algorithm in terms of runtime and accuracy. Yoon et al. [10] proposed an algorithm called net-effect-based stream outlier detection based on a set-based method for global outlier detection to make the detection of distance-based outliers extremely fast. Yuan et al. [11] proposed weighted-frequent-pattern-based outlier detection, which consists of two phases, weighted frequent pattern mining and global outlier detection. A synthetic dataset was used, and they analyzed the efficiency of the proposed method in finding frequent pattern outlier factors, longer frequent patterns, and online outliers for high-dimensional time series based on maximal frequent pattern methods with sliding windows of different sizes.

Alghushairy et al. [12] proposed the genetic-based incremental local outlier factor for global outlier detection. Four datasets were selected, and the results of the proposed method were compared to those of the density incremental local outlier factor algorithm. Alsini et al. [13] improved the grid-partition-based local outlier factor algorithm (GP-LOF) to detect global outliers. The authors named the new algorithm the grid-partition-based local outlier factor based on reachability distance. The University of California Irvine vowel dataset, KDD99 SMTP dataset, and University of California Irvine shuttle dataset were selected to test the proposed algorithm, and the results were compared to those of the GP-LOF algorithm. Gao et al. [14] proposed the cube-based outlier detection algorithm for global outlier detection. Experiments were conducted using five datasets, and the

results were compared to the incremental local outlier factor algorithm in terms of accuracy, average runtime, and average runtime memory. Anomaly detection with a sparsity profile was proposed to detect global outliers in Moon et al. [15]. Accuracy, approximation ratio, and error rate were used to evaluate the proposed model against the local outlier factor algorithm and the autoencoder.

Yu et al. [16] suggested the compressed sensing and online extreme learning machine autoencoder algorithm for global outlier detection. Zhu et al. [17] proposed the grid-based approximate average outlier detection framework, which was based on a min-heap-based algorithm, k-skyband-based algorithm, and grid-based index to detect global outliers. Gruhl et al. [18] suggested an online extension for global outlier detection using an indegree number technique based on the K-nearest-neighbor-based outlier detection method using an indegree number to detect outliers. A total of 14 datasets were selected for the experiment, and the results of the proposed method were compared to those achieved with the half-space trees, isolation forest, and local outlier factor algorithms in terms of average F1-score. Togbe et al. [19] extended isolation forest anomaly detection in the data stream algorithm for global outlier detection. Three real datasets were used, and the results of the proposed method were evaluated in terms of running time, F1-score, and memory consumption. Wang et al. [20] proposed multiple random convolution kernels, and robust random cut forest to detect global outliers. The proposed method was run on five datasets, and the results were evaluated in terms of precision, recall, area under the receiver operating characteristic curve and F1-score.

Zhao et al. [21] extended two layers of the robust anomaly detector framework with ensemble prediction to deal with global outliers. Three datasets were collected from the cluster, Internet of Things platform, and real celebrity images, and the accuracy of the proposed method was compared with that of the other algorithms using a celebrity images dataset. Ariyaluran Habeeb et al. [22] proposed a framework to detect global outliers and developed a streaming sliding window local outlier factor coreset clustering algorithm. Three datasets were used to examine the proposed framework, namely the Defense Advanced Research Projects Agency intrusion detection dataset, the Mid-Atlantic Collegiate Cyber Defense Competition dataset, and the dataset from DEFCON21, the world's largest hacking conference, which is held annually. The memory consumption, accuracy, and execution time of the proposed framework were compared to those of existing methods. Jiang et al. [23] proposed an algorithm for global outlier detection called classification characteristics of marine data based on the density-based spatial clustering of applications with a noise algorithm in batch processing. Data were collected from the Argo Real-Time Data Center. True-positive results, false-positive results, and the Davies-Bouldin Index were selected to compare the results of the proposed algorithm with those of density-based spatial clustering of applications with noise and the K-means algorithms.

Benjelloun et al. [24] suggested an algorithm based on improved micro-cluster-based outlier detection (MCOD), Abstract-C, and Exact-Storm algorithms to detect global outliers in high-dimensional streams. Two datasets were collected from the University of California Irvine Machine Learning Repository, namely the Wisconsin Breast Cancer (WBC) dataset and the SpamBase dataset. For the evaluation, the proposed algorithm was compared with the existing algorithms MCOD, E.Storm, and Abstract-C, which had been enhanced based on lifecycle status. The hybrid model achieved a maximum accuracy of 92.42, a recall of 99.17, a precision of 82.41, and an F-measure of 90.02 on the WBC dataset, and the hybrid model achieved a maximum accuracy of 97.89, a recall of 54.55, a precision of 70.59, and an F-measure of 61.54 on the SpamBase dataset. Xu et al. [25] improved the local outlier factor algorithm to detect global outliers in batch processing.

Najib et al. [26] proposed the fuzzy c-means clustering framework to deal with missing values. In the evaluation, the results were compared with those of existing algorithms in terms of the average confusion matrix, average F-score, and average specificity with different rates of missing values. Shen et al. [27] proposed a flexible ensemble algorithm based on a soft voting approach on Spark to tackle the problems of outliers, duplication,

and missing values. The algorithm first deduplicated the data, and then set the missing values to −2. Then, the −2 values and the outliers were replaced with the null value, and then the null value was recovered using linear regression. Four datasets, namely Satellite, BreastW, Shuttle, and HTTP, with a different percentage of outliers (32%, 35%, 7%, and 0.49%, respectively) were utilized. The Shuttle and HTTP datasets achieved 96% precision, whereas the Satellite dataset had the lowest precision at 90%. The HTTP dataset had the highest recall at 99%, whereas the Satellite dataset had the lowest at 82%. A real-world dataset from a region in Southwest China was selected as an examination dataset to test the proposed method's ability to detect outliers, and the results were compared with those of two algorithms' density-based clustering of applications with noise and a local outlier factor. The proposed algorithm had the highest precision at 99.1% and highest recall at 95.9%, while the local outlier factor algorithm had the lowest result with 95.6% precision and recall.

Lizhen et al. [28] adopted the neighbor sorting method to deal with duplication in batch processing. The data which were utilized to evaluate the proposed method were collected from the transformer station in Baiyin and Belgium's national Distribution Network (operation data and corresponding weather data). The results achieved by the proposed method were presented in figures. Liu et al. [29] improved the cluster-based local outlier factor (CLOF) and the random forest algorithm to impute missing data and detect outliers in batch processing. Datasets comprising data on electricity consumption, network loss rate, and active power were utilized to compare the results with those achieved by the old CLOF algorithm in terms of detecting outliers, and they were also compared to K-nearest neighbors, MICE, and missForest results in terms of imputing missing values. The improved CLOF detected the highest number of outliers in the three datasets and achieved the highest recall of 100 on the active power dataset and the network loss dataset and a precision of 88.33 and 92.31 on the datasets, respectively. However, the original CLOF achieved a higher precision, 95.69, on the electricity consumption dataset. In imputation data, K-nearest neighbors had the highest performance in the majority of imputation results.

Various techniques based on a combination of different approaches in machine learning have been suggested to tackle data cleaning issues. For example, Thakur et al. [30] combined K-means and the C5.0 decision tree to detect global outliers in batch processing. Data related to credit card transactions were used, and the accuracy, runtime, and performance of the proposed method were compared to those of existing methods. Heigl et al. [31] integrated the classic isolation forest and extended isolation forest to detect global outliers. Four datasets, HTTP, SMTP, Forest Cover, and Shuttle, were selected for the experiment and compared to iForest anomaly stream detection and growing random trees. Rivera et al. [32] combined K-means unsupervised learning and K-nearest neighbors supervised learning for global outlier detection.

Degirmenci et al. [33] merged the local outlier factor and density-based spatial clustering of applications with noise for global outlier detection. A total of 16 datasets were used for the evaluation, and the results of the proposed algorithm were compared with those of existing algorithms in terms of area under the receiver operating characteristic curve analysis and statistical tests. Panneerselvam et al. [34] combined supervised and unsupervised machine learning to detect global outliers in data batches. The data were collected from smart meters installed in Ireland. The proposed model was evaluated based on accuracy, precision, recall, and F1-score, which were determined to be 90%, 87.5%, 91.3%, and 89.4%, respectively. Prabhakar et al. [35] combined an adaptive deep belief network with a deer hunting optimization algorithm for global outlier detection in data batches. Data were collected from call detail records, and the proposed algorithms were compared to existing methods.

Pei et al. [36] combined a spark framework with a support vector machine and random forest algorithm for global outlier detection. For the experiment, the KDD CUP99 dataset and NS_KDD dataset were used and compared to a traditional vector machine in terms of detection efficiency. The proposed method had a higher detection efficiency than the

traditional vector machine in both experiments; in the KDD CUP99 dataset, the proposed method had 98.3% efficiency, whereas the traditional vector machine had 92.1%, while in NS_KDD dataset, the suggested approach had 99.4% efficiency, while the traditional vector machine had 81.4%. Xu [37] combined association rules with density-based spatial clustering of applications with noise and also improved the performance of the correlation vector machine (RVM) in detecting global outliers. The data were collected from turbine flue gas, and the prediction result was compared between the RVM and improved RVM in terms of absolute error and prediction time. The results showed that the improved RVM had a shorter prediction time and smaller absolute error compared with the RVM. Andreoni Lopez et al. [38] merged the normalization algorithm and feature selection algorithm to deal with irrelevant data and GTA; the Federal University of Rio de Janeiro dataset was used, and this method was compared with principal component analysis, sequential feature selection, support vector machine recursive feature elimination, and ReliefF methods in terms of accuracy and sensitivity of detection using three classification methods, namely, decision tree, support vector machine, and neural network. The principal component analysis achieved the best results in all classifications in terms of accuracy and sensitivity of detection.

3.3. Deep Learning

Numerous studies have proposed deep learning approaches to address data cleaning issues, such as one conducted by Zhang et al. [39], who improved density-based clustering of applications with noise density clustering to deal with duplicated data in batch processing. Authoritative data from the University of California were collected to analyze the proposed method. The results were as follows: the recovery rate oscillated, the correct detection rate increased, and the accuracy decreased when the number of record increased. Fitters et al. [40] proposed an outlier-enriched long short-term memory framework to detect contextual outliers. Data were collected from the urban traffic network of Hague, and the results were compared to those resulting from naïve multivariate long short-term memory in terms of running time. Arora et al. [41] suggested a framework to detect global outliers consisting of five stages, namely data collection, data pre-processing, data splitting, optimization and training models, and model evaluation using deep learning, including long short-term memory and a recurrent neural network for the offline mode and deep neural networks for the online mode. PubNub sensor datasets were used for the evaluation, wherein the offline mode was compared among different models and weather sensor data were used to evaluate the online model.

Iturria et al. [42] proposed an algorithm named the ensemble of online recurrent extreme learning machines anomaly detector based on a neural network. Six time series datasets were used for the experiment, and the results were compared to those of existing methods in terms of precision, recall, F-measure, and NAB scores for detecting global outliers. Wang et al. [43] proposed a method based on an encoder–decoder reconstruction model to detect global outliers in batch processing. The data were collected from diesel engine assembly data and the result was compared to that of the original encoder and decoder model. Zhou et al. [44] suggested an online detection method based on multi-scale deep learning for global outliers. The data that were used were collected from the University of California Riverside archive. The results were compared to those of the random forest, support vector machine, k-nearest neighbors, decision tree, logistic regression, gradient boosting, and Gaussian naïve Bayes algorithm in terms of accuracy. The proposed method had 95% accuracy, which was the highest among all algorithms.

In contrast, many studies have integrated diverse approaches in deep learning to deal with data cleaning issues. Albattah et al. [45] combined convolutional neural networks and long short-term memory techniques (hybrid ConvLSTM deep learning) to detect global and contextual outliers. The Multiple Intelligent Monitoring in Intensive Care dataset was used, and the proposed method was compared to convolutional and long short-term memory in deep learning and linear regression, decision tree, random forest, and support vector

machine methods. Belacel et al. [46] combined long short-term memory and recurrent neural network encoder–decoder architecture to detect global outliers. Eight datasets from the University of California Irvine repository were utilized. Area under the curve, area under the precision–recall curve, weighted average F1 score, and time were used to compare the results of the proposed method with those of existing methods. Gao et al. [47] merged unsupervised deep learning and shallow feature fusion learning to detect global outliers. Data collected from an electricity company in China were used to compare the proposed method with other outlier detection models, namely deep belief networks including the one-class support vector machine, cost-sensitive deep neural network ensemble, variational-autoencoder-based control chart, one-class support vector machine, concatenated features + SMOTE-Tomek Random Forest + one-class support vector machine, generative-adversarial-network-based feature reconstruction, and generative-adversarial-network-based data generation in terms of outlier detection rates. The results show that the proposed model achieved the highest detection rate at 99.7 compared to the other models.

3.4. Statistical Techniques

Multiple studies used statistical techniques to investigate data cleaning issues. Smrithy et al. [48] suggested non-parametric statistical techniques to detect global outliers, and the results of the proposed algorithm were compared to those of other algorithms in terms of metrics (true positive, true negative, false positive, and false positive). Yu et al. [49] proposed the cumulative kernel density estimator with a retrospect algorithm to detect global outliers. Two real-world datasets from the University of California Irvine (KDDCup99, CoverType) and two synthetic datasets were used for the experiment, and the results were compared to the stream outlier detector_graphics processing units method in terms of F-score, precision, and recall. Karn et al. [50] suggested statistical methods for detecting global outliers using the log-likelihood ratio. The dataset was collected from JSON logs, and the method was evaluated in terms of precision, recall, and F1-measure using different thresholds.

Jamshidi et al. [51] proposed three algorithms to detect global outliers, the standard Z-score, the modified Z-score with decomposition, and the exponential moving average in batch processing. Data were collected from surface water temperature datasets, and the results of these algorithms were compared in terms of precision, recall, and F-score. Kurt et al. [52] suggested computing a group of univariate summary statistics to deal with global outliers. Data from the Human Activities and Postural Transitions dataset were collected from the University of California Irvine repository, and the average detection rate achieved was compared to that achieved with existing algorithms. Bobulski et al. [53] modified the moving average series algorithm to detect global outliers in batch processing. Data pertaining to temperature readings were collected from IoT devices for the experiment to compare the performance of the moving average algorithm and the proposed moving average algorithm using the signal-to-noise ratio as the parameter. The experiment results show that the proposed moving average error correction algorithm performed marginally better than the classic moving average algorithm and slightly improved the value of the signal-to-ratio coefficient.

Kulanuwat et al. [54] used a sliding window to develop a median-based statistical outlier detection method to tackle missing values and global outliers. Time series data of water levels were collected from a telemetry station. Extensive experiments were conducted using statistical methods for automated anomaly detection and several data filling methods, with the results showing that median-based statistical methods for anomaly detection combined with the linear and the spline method for data filling yielded promising results for non-cyclical data behaviors. Fountas et al. [55] proposed a data imputation model based on data from various IoT devices and adopted a continuous correlation detection methodology to deal with missing values. Zhao et al. [56] proposed variational-based imputation for multi-modal time series. For better convergence, a two-stage isolated optimization strategy was also proposed.

Bimonte et al. [57] suggested a linear-programming-based framework for batch processing to handle missing data in multi-granular data warehouses. The datasets used for the experiment were the FoodMart dataset and the Car Evaluation dataset. Fang [58] proposed the diversity-based sample selection (SDUS) algorithm based on the correlation of redundant data to deal with missing and duplication in batch processing. The dataset was gathered from the IoT, and the accuracy of the proposed algorithm was compared with that of the ST-SDC algorithm. The results showed that SDUS achieved 96% accuracy, whereas the ST-SDC algorithm had an average accuracy of 87%. Jehlol et al. [59] proposed a method based on Pearson correlation between histograms of various data extensions to remove duplicated data. They used 3DLDF files and SQLite files for the evaluation, and the results were compared to the two thresholds two divisors and basic sliding window methods in terms of data size after removing duplicated data, and percentage of removed duplicated data. The proposed method had better results than other methods in both experiments.

3.5. Combined Techniques

Several researchers have developed novel methods by merging the aforementioned techniques, for example, machine learning with deep learning or statistical and machine learning. For example, Xiao et al. [60] combined a probability-based algorithm and a clustering algorithm to detect global outliers in data batches. Household electricity data were used to evaluate the proposed method by comparing it with existing methods in terms of running time, precision, and accuracy. Sun et al. [61] merged locality-sensitive hashing and the dynamic isolation forest to detect global outliers. The occupancy dataset and Buzz dataset were used for the experiment, and the results were compared to those achieved with the locality-sensitive hashing isolation forest and Wadjet in terms of F1-score, time, and the area under the curve. Reunanen et al. [62] combined logistic regression, stochastic gradient descent, and hidden representation to predict outliers and used a hidden layer of neurons and autoencoders to detect global outliers. The proposed method was applied to sensor data. The result was compared with that of Storme2, continuous outlier detection, and micro-cluster-based continuous outlier detection algorithms in terms of recall, the false-positive rate, the receiver operating characteristics curve, and the area under receiver operating characteristics curve. To improve global outlier detection, Crépey et al. [63] combined principal component analysis and neural networks. A synthetic dataset and a real dataset from a financial data provider were used, and the results were compared to those achieved with the isolation forest, local outlier factor, density-based clustering of applications with noise, K-nearest neighbors, support vector machine, and sig-IF (combination of feature extraction step through a path signature computation with isolation forest) algorithms in terms of accuracy, F1-score, precision, and recall. Huang et al. [64] utilized maximum likelihood estimation based on particle swarm optimization and used a sea level dataset to compare the proposed method with existing methods for global outlier detection.

Surapaneni et al. [65] used a dense stream algorithm and gated-recurrent-unit-based recurrent neural networks to detect global outliers. Data on network traffic details were collected from Cisco, and the accuracy of the proposed algorithm was compared to that of the DenStream algorithm. Zhang et al. [66] proposed two parallel pipelines using deep learning to detect global outliers, and the proposed method was applied to two datasets that were collected from an e-commerce company. The training loss and F1-score of the proposed method were compared to those of other methods that use an intelligent baseline module or unsupervised detection module. García-Gil et al. [67] proposed two algorithms, a homogeneous ensemble filter and a heterogeneous ensemble filter, for global outlier detection in batch processing. Four datasets, SUSY, HIGGS, Epsilon, and Evaluation Computation for Big Data and Big Learning 14, were used for the evaluation, and the results in terms of accuracy and running time with different levels of noise were compared to the those of the edited nearest neighbor for the big data algorithm using two classifiers, namely decision tree and K-nearest neighbors. The results show that the proposed framework can

successfully deal with noise. Ma et al. [68] combined long short-term memory with bi-directional imputation and transfer learning to deal with missing values in batch processing, using energy consumption data from facilities and campus services at Cornell University. The results were compared to those of existing methods.

3.6. Unclassified Techniques

Various techniques have also been proposed that cannot be classified into one of the aforementioned classifications, so we refer to these techniques as unclassified techniques.

Li et al. [69] combined a sliding window and control chart called the double cumulative sum based on a data stream to improve the accuracy of global outlier detection. The proposed method was compared to automatic outlier detection for data streams and sliding nest window chart anomaly detection based on data stream algorithms in terms of runtime with windows of different sizes and areas under the curve. Rollo et al. [70] integrated an outlier classifier and filtering technique to detect global outliers by using data gathered from the traffic sensor network of the city of Modena. The results of the proposed method were presented in plots. Zhu et al. [71] combined three modules based on the edge end, where the first module was used for global outlier detection, the second module was used for data reduction, and the last module was used for outlier classification. Two datasets, the Cyclone Wildfire Flood Earthquake Database and the Comprehensive Disaster Dataset, were used, and the results of the proposed module were compared to those of existing methods in terms of data compression and accuracy. Yang et al. [72] proposed the filtering–identifying-based anomaly detection algorithm, which comprised two complementary methods, the first of which combined identifying and filtering methods, and the second of which combined data-oriented general and data-oriented specific methods for global outlier detection. Data were collected from a ground-based wide-angle camera on the third day of the experiment using a catalog stream dataset and catalog stream located dataset, and the results were compared to those achieved with drift spot, long short-term memory nonparametric dynamic thresholding, normalized feature deviation, and Wavelet in terms of F1-score.

Amen et al. [73] proposed a distributed collective anomaly detection method to detect collective outliers. The dataset was collected from an IoT project (Highway Road Traffic Monitoring System). The experiment result of the proposed method was compared to that achieved with the local outlier factor and adaptive stream projected outlier detector algorithms in terms of accuracy. The proposed method had a maximum accuracy of 75%, which was much higher than that achieved with the local outlier factor and adaptive stream projected outlier detector algorithms. Chen et al. [74] suggested a framework based on the randomized principal score and generative principal score (randomized algorithms) to detect global outliers in large streaming data with differing correlation strengths. A large server log dataset and a US stock daily price dataset were used to compare the proposed method with existing methods. The results show that the proposed approach improves computation efficiency and scalability for principal score calculation. Manjunatha et al. [75] proposed a dynamic mechanism to detect global outliers. For evaluation, three case studies from social network data were used and the results of the proposed method were evaluated in terms of F-score, precision, and recall.

Su et al. [76] proposed abnormal conditions detection using the continuously monitored objects approach for global outlier detection. The tropical atmosphere ocean dataset and the generated dataset were utilized for the experiment, and the results were compared with those achieved using the incremental local outlier factor algorithm in terms of time requirements, space requirements, and F-score. Cao et al. [77] suggested a lightweight method and approximate trajectory outlier detection using the trajectory data stream algorithm to detect global outliers, and the proposed method was compared to trajectory outlier detection using trajectory data streams and isolation-based anomalous trajectory algorithms in terms of running time and accuracy. Dias et al. [78] achieved a decreased anomaly score by using a repeated sequence algorithm to detect global outliers. It was

evaluated in terms of running time, memory usage, false-positive rate, and false-negative rate, and compared with several outlier detection methods.

Borah et al. [79] proposed a parallel outlier detection algorithm for detecting contextual outliers in data streams using a graphics processing unit. For the experiment, four real datasets from the University of California Irvine repository and a set of synthetic data streams were used, and the results were evaluated in terms of accuracy and average processing time in milliseconds. The best results achieved were as follows: 95% precision, 99% recall, and a 97% F-score in 37.8 ms.

Dani et al. [80] proposed an online outlier detection algorithm using recursive residuals via the recursive least-squares method for global outlier detection. A simulation was conducted on historical data for Tesla stock prices for a given period, the results of which show that the proposed online algorithm achieved high accuracy for anomaly detection at a low computational cost. Leigh et al. [81] suggested a 10-step framework to detect global outliers in batch processing using high-frequency water quality data from in situ sensors, reflecting turbidity, conductivity, and river level, collected from rivers flowing into the Great Barrier Reef, and the framework was compared with existing methods.

Souza et al. [82] proposed a new method for detecting global outliers based on the combination of a multiway decomposition technique with multivariate techniques to identify patterns in data that were collected from smart urban sensors. Their three-stage method involved dimensionality reduction, classification of latent factors, and a combination of these two to generate an urban space pattern identification model. The results show that the proposed tensor decomposition technique improves outlier detection and provides useful information for enhancing city planning and operation. Gupta et al. [83] proposed a framework for online error detection to enhance the reliability of data analytics. The two datasets that were used to evaluate the performance of the framework were a real-time air quality dataset and an Intel sensor dataset. The performance of the framework was illustrated in terms of the false-positive rate and percentage of error in a dataset; when the error percentage increased in the dataset, the false-positive rate increased too. Zheng et al. [84] proposed a data quality identification model (tri-training) to detect global outliers in batch processing. A power consumption dataset collected from various industries was used to evaluate the proposed model, and the results were compared to those achieved with existing methods in terms of error rates. Wang et al. [85] applied an angle-based outlier at an edge node to detect global outliers, using data gathered from sensors, and its performance was compared with that of existing methods.

You et al. [86] dealt with duplication and irrelevant data by proposing an online streaming feature selection window algorithm with a sliding window strategy. For the evaluation, 14 datasets were used, and the accuracy and runtime were compared with alpha investing, online streaming feature selection, and scalable and accurate online approach algorithms. Pezoulas et al. [87] dealt with missing values, duplication, and outliers in batch processing by proposing a three-phase framework using data from the University of Athens and Harokopio University of Athens. The results of the proposed method were presented. Salloum et al. [88] proposed the random sample partition (RSP-Explore) method to tackle statistical estimation, detect errors, and clean data in batch processing by exploring big data on small computing clusters consisting of five nodes. The proposed method was evaluated on three datasets, namely HIGGS data, power data, and airline data. Numerical data were used in this experiment, and the proposed method was not designed to be applied to data streaming. The statistics estimator computed univariate and bivariate correlation, whereas the error detector classified the data into four groups: missing values, outliers, errors, and valid values. Ju et al. [89] proposed an optimization technique based on task merging to detect duplicated data in batch processing. Datasets were collected from the Zhengzhou Commodity Exchange, and artificial datasets were collected from the transaction processing performance council, and the results were compared with those of the existing methods.

Ding et al. [90] proposed an updating algorithm combining data cleaning and the conjugate gradient to deal with missing values in batch processing. The data used for the

experiment were from the US Census 1990 and the Thyroid Medical datasets. The results compared precision and recall in the Thyroid Medical Dataset using a logical regression model, and the accuracy in the US census dataset was compared to that in the updating algorithm via random sampling. The proposed algorithm showed a better performance than the logical regression model. Rama Satish et al. [91] proposed a hybrid algorithm by combining cuckoo search optimization with the gravitational search algorithm to tackle missing values, duplicated data, and outliers in batch processing. An employee dataset was used to examine the performance of the proposed algorithm in terms of precision, recall, accuracy, and F-measure and the results were compared with an existing method for cleaning data without optimization. Figure 5 shows the relationship between techniques and data cleaning issues, where the nodes represent different studies, the existence of a node for a technique is shown, and the data cleaning issue category indicates which technique studies used to solve this issue. We uploaded a table summarizing previous studies online due to the size of the table (the table summarizing the literature review can be found at https://latrobeuni-my.sharepoint.com/:w:/g/personal/18048071_students_ltu_edu_au/EZZaU6-QmYdJsEBfQRg3XqcBxhcjwrGF7rRqOLbOeOGOBw, accessed 1 July 2023).

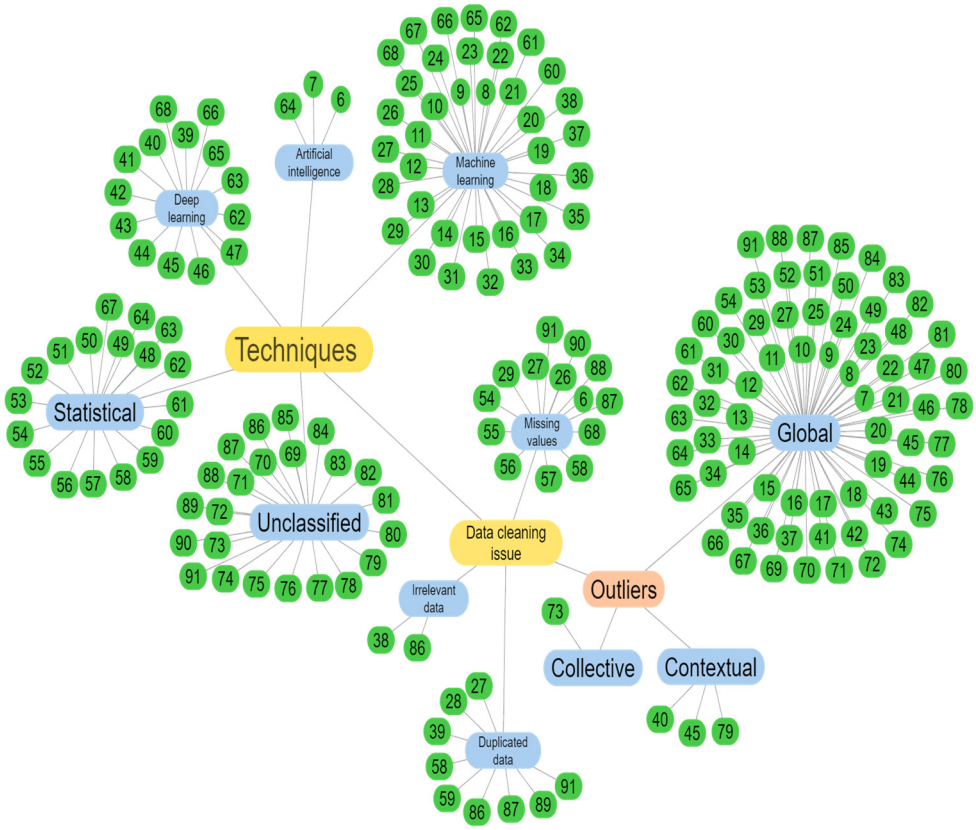

Figure 5. Correlation between techniques and data cleaning issues.

4. Discussion

In this section, the findings from the previous section are classified based on the SLR questions as data-cleaning operations, techniques, evaluation methods, and future directions and challenges.

4.1. RQ1: Why Is It Important to Clean Data Streams?

The number of studies on data cleaning is continually increasing, which indicates the importance of improving data quality as this is a key factor in obtaining accurate data analysis results, as shown in Figure 6.

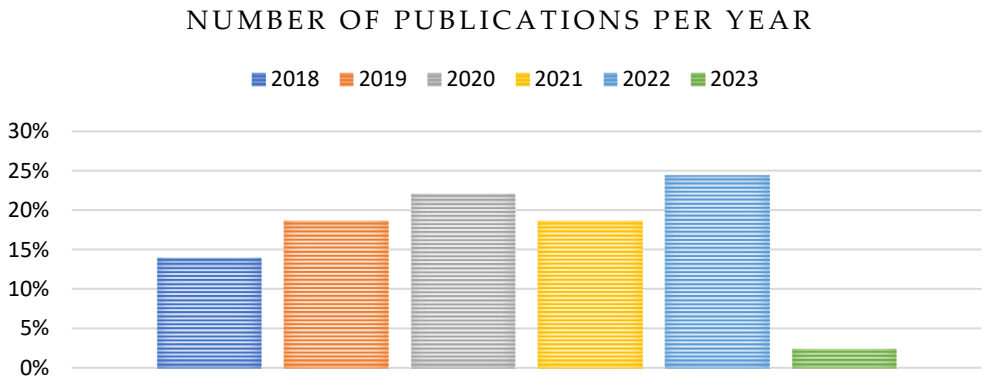

Figure 6. Number of publications each year for data cleaning.

The figure above indicates that the number of studies on data cleaning has increased each year. However, the number of articles published in 2023 is low because this SLR was conducted at the end of February in 2023, so there is a possibility more studies about data cleaning were published this year, in accordance with the trend seen in prior years. Improved data quality leads to more accurate and trustworthy results, and data quality depends on how thoroughly data have been cleaned. There are two types of data processing, batch processing and stream processing. The number of studies on streaming data has increased over the past few years, likely due to the increased use of technologies and the internet around the world, generating massive data. The number of studies on batch processing, however, remained relatively constant from 2018 to 2020. Cleaning data streams is important because an issue can be detected in the initial stage, so a decision regarding this issue should be made before it becomes a bigger and more costly issue to address. Figure 7 presents a comparison between the data processing types.

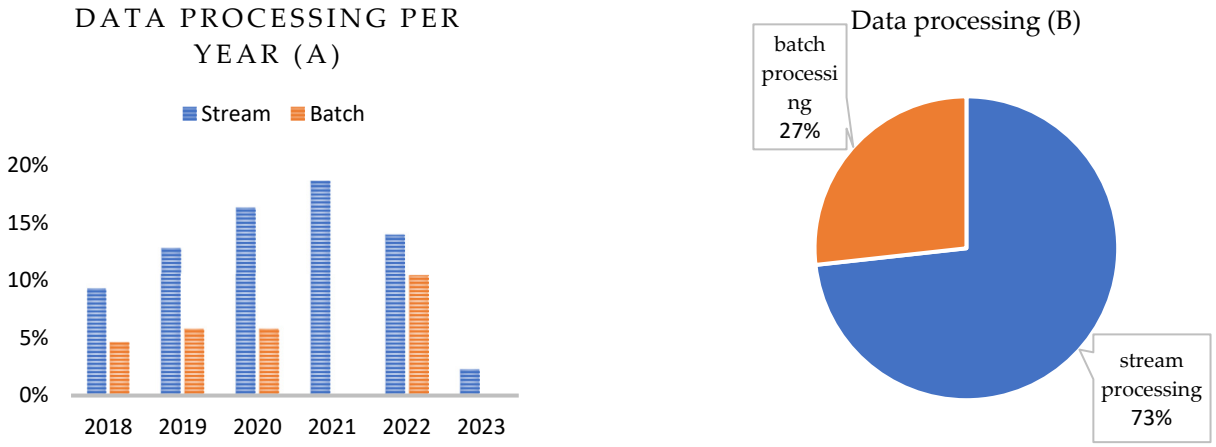

Figure 7. (**A**) shows that the number of studies on stream data cleaning has been higher than those on batch data cleaning for the past five years, which reflects the importance of data stream cleaning. (**B**) demonstrates that most of the techniques proposed in the literature related to stream processing, with 73% of the total studies in the last five years examining streaming data, whereas 29% focused on batch processing.

4.2. RQ2: Which Data Cleaning Issue is Most Commonly Discussed during the Data Cleaning Process?

Several issues, such as missing values, duplicate, outliers, and irrelevant data, tend to appear during data cleaning processing, and many approaches have been proposed to deal with these.

According to the bar chart in Figure 8, around 73% of the research on data cleaning issues is devoted to the detection of outliers, and this number has increased each year

from 2018 to 2023. This reflects the significant effect of outliers on data analysis results and the importance of detecting outliers to improve data quality. The pie chart indicates that 95% of the total outlier detection studies focused on detecting global outliers, whereas 3% focused on contextual outliers. Furthermore, 2% of the studies focused on collective outliers compared to other types of outliers. Referring to the bar chart in the figure above, almost 13% of existing studies tackled more than one issue, such as missing values, outliers, and duplicated data, whilst approximately 8% of the studies investigated missing values in data cleaning, and around 5% of the studies conducted focused on removing duplicated data.

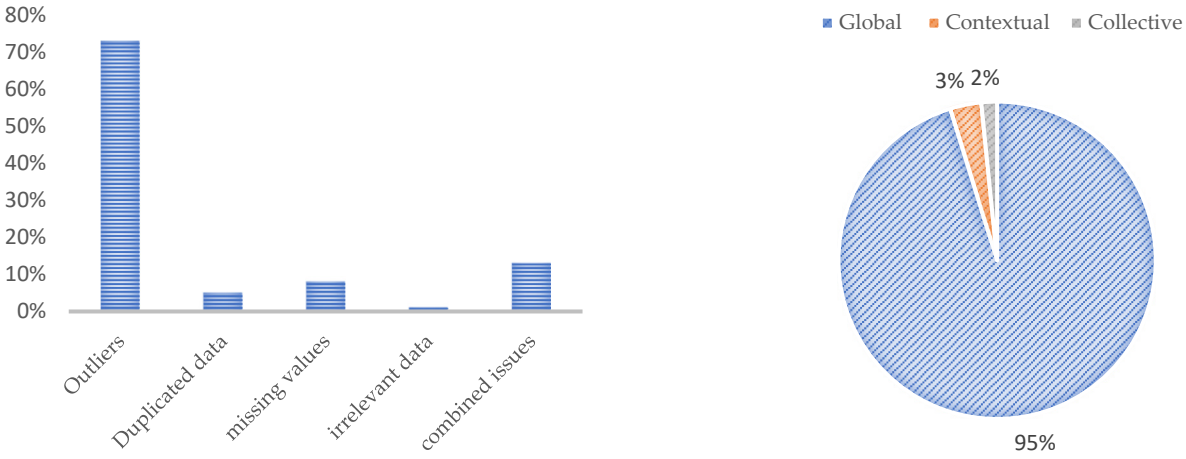

Figure 8. Tackled data cleaning issues.

Fewer studies have been conducted on identifying irrelevant data, one possible reason for this being that irrelevant data are difficult to identify except in specific cases where the irrelevant data can be identified. For example, if data analysis was to be conducted to evaluate the general health of a population, phone numbers would not be relevant to the results.

Although most research conducted in the last five years has focused on detecting outliers, the logic sequence may help to improve the accuracy of outlier detection. For example, Amen et al. [73] dealt with collective outliers without deleting redundant data, so redundant data should be removed prior to detecting outliers, particularly collective outliers, to improve the accuracy of collective outlier detection. Collective outliers are a group of data that deviate significantly from the entire dataset, where a single point is probably considered as normal data Han et al. [3]. Thus, duplicated data could be considered as a collective outlier, which would increase the false-positive rate. Furthermore, both Fitters et al. [40] and Albattah et al. [45] detected contextual outliers without tackling missing data. In addition, Liu et al. [29] and Kulanuwat et al. [54] both dealt with outliers before dealing with the missing value. However, missing values need to be tackled before detecting outliers, especially contextual outliers, because missing values may hide outliers and affect outlier detection Han et al. [3].

Logic sequences are also needed when dealing with several issues; for example, removing duplicated data should be carried out prior to dealing with missing values, and both duplicated data and incomplete data need to be handled before detecting outliers; this is similar to what was described in Shen et al. [27], in that even though the sequence was logically right, the implementation was not because it was re-imputing the value of missing values and outliers after a value of -2 was assigned to the missing data in the initial stage and outliers were detected.

4.3. RQ3: What Sort of Techniques are Commonly Used to Clean Data?

Many techniques have been applied to data cleaning to improve data quality, the most commonly used techniques being machine learning techniques. The majority of data

cleaning research has been conducted on machine learning techniques, with 36% of the total studies in the last five years focused on these, followed by research on statistical techniques at 14% and research on deep learning techniques and combined techniques at 10% each. Finally, 2% of previous studies were conducted using an artificial intelligence technique to clean data. Figure 9 presents a graph detailing the total usage of all techniques for data cleaning.

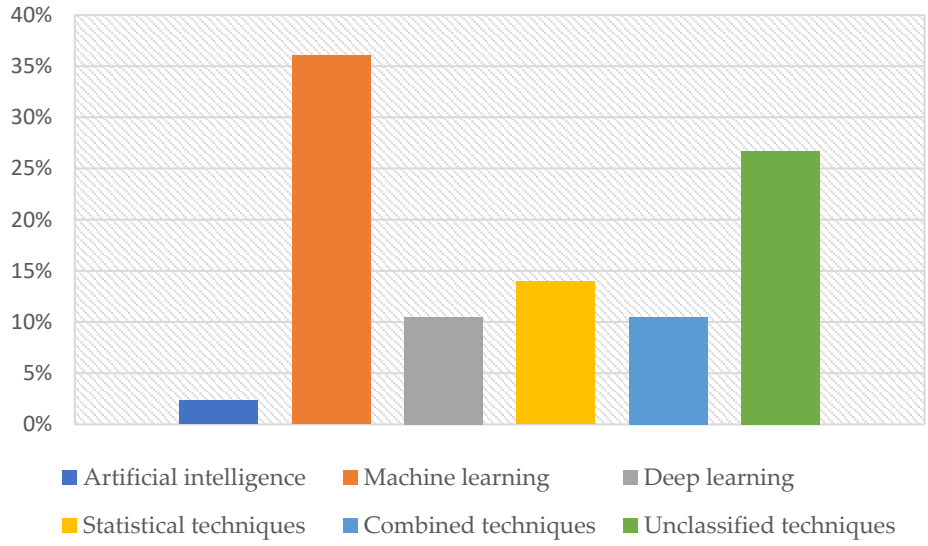

Figure 9. Techniques used for data cleaning issues.

4.4. RQ4: What Methods Have Been Used to Evaluate the Proposed Approaches?

Researchers have used many approaches to evaluate the effectiveness of their proposed techniques. The effectiveness of the proposed techniques can be examined in several ways, such as by using metrics, comparing the proposed approach with other approaches, or running the proposed method and presenting the results. Figure 10 shows the percentage of studies using these evaluation methods.

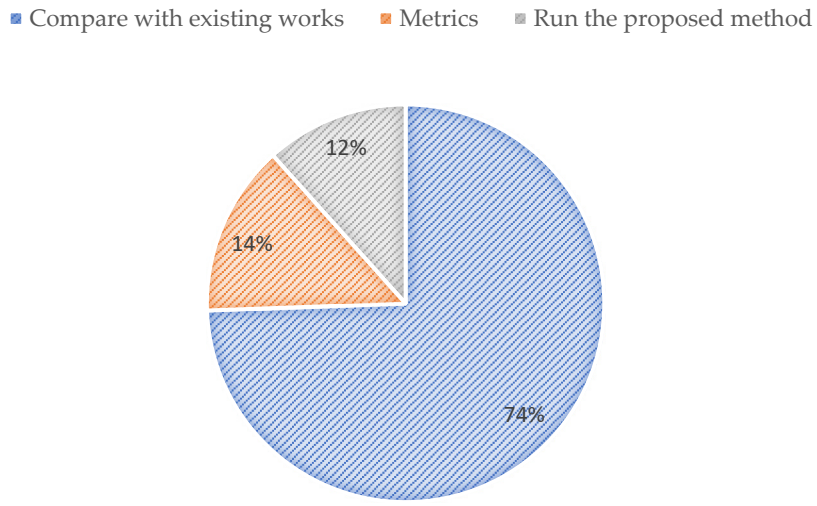

Figure 10. Percentage of studies using various evaluation methods.

A total of 74% of previous studies compared their proposed method with existing methods using running time and metrics like precision, recall, accuracy, and F1 score. For instance, Ma et al. [68] and Yang et al. [72] compared their method with several methods, like linear regression, K-nearest neighbors, and recurrent neural networks, to

validate the effectiveness of proposed method in dealing with missing values. Only 14% of studies evaluated the suggested method by using metrics without a comparison with other methods. For example, Wang et al. [20] used metrics such as recall, precision, F1 score, and area under receiver operating characteristic curve to evaluate the effectiveness of the proposed approach in detecting global outliers. Another way to evaluate such methods is running the proposed techniques on a dataset and comparing the results from before and after applying the proposed technique; for instance, in Dani et al. [80], the proposed method was run on the Tesla dataset to detect a global outlier, the results were presented in a scatter plot as validation of the suggested approach's effectiveness. Therefore, comparing the obtained results with those of other existing techniques is an effective approach that researchers use to illustrate the efficiency of their proposed method.

4.5. RQ5: What Are the Future Directions for Data Cleaning?

We identify possible future directions in data cleaning and the challenges that need to be tackled and the methodology that has been used to deal with these issues. Furthermore, we believe that the sequence of the data cleaning process also plays a key role in increasing the accuracy of data cleaning and reducing execution time. The challenges of cleaning data are classified as follows.

4.5.1. Nature of the Data

The nature of the data, namely whether they are structured, semi-structured, or unstructured, presents a challenge when cleaning data. It is particularly challenging to clean semi-structured and unstructured data.

4.5.2. Outliers

The majority of research conducted in the last five years on detecting outliers has focused on the detection of global outliers rather than contextual and collective outliers, so more research on detecting contextual and collective outliers is needed. Also, none of the previous work suggests a method for detecting all three types of outliers for either data streams or batch data.

4.5.3. Duplicated Data

Scant research has been conducted on the removal of duplicated data, especially in big data streams. Removing duplicated data in streaming data would improve the accuracy of detecting outliers, especially collective outliers. However, deleting duplicated data is not easy, particularly in big streaming data in different data formats.

4.5.4. Missing Values

Researchers have used three methods to deal with missing values, namely ignoring, imputing, and deleting missing data. There is a connection between detecting outliers and imputing missing values since imputing missing values improves the precision of detecting outliers, especially contextual outliers. However, identifying missing values in big data streams which have semi-structured and unstructured formats is difficult.

4.5.5. Windowing

Windowing allows for more control over data, especially data streams. Figure 11 illustrates the percentage of previous studies that used window techniques in their proposal approach for data cleaning, showing that nearly 52% of former works used windowing in their experiments. The majority used sliding windows, and only one used polling windows. Thus, combining different types of windows has not yet been used to clean big data streams.

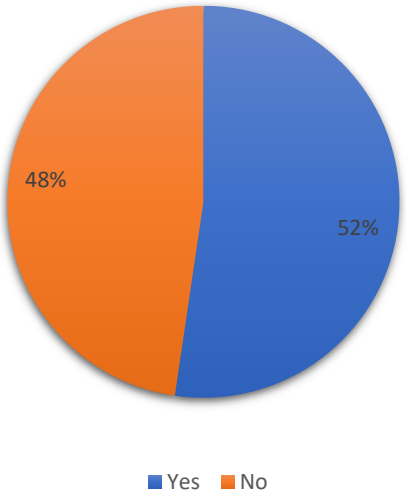

Figure 11. Usage of window techniques.

4.5.6. Framework

Using a framework in cleaning data streams would help to improve the data stream quality by allowing for the easy detection of various data issues in the data stream and the selection of an appropriate action to deal with it, because several functions can be applied to the framework. Thus, designing a framework that integrates data cleaning and data programming needs more investigation.

5. Conclusions

In this paper, we presented a brief introduction to big data and the characteristics of big data which make it different from traditional data. Furthermore, we discussed the differences between batch processing and streaming processing and explained the different types of data analysis. We narrowed our search to the data pre-processing phase, the step before data analysis, and we presented four issues that may appear during data cleaning operations and gave an example of each issue. We also discussed the related work from the last five years, and these were classified based on the data cleaning issues they addressed. We detailed five research questions, and the SLR strategy was presented. These questions were designed to give an idea of the significance of data cleaning, the techniques that have been used to date for data cleaning, and the effective evaluation methods that can be used to assess the efficiency of the proposed approach. Finally, we suggested future research directions and discussed the big streaming data challenges in relation to data cleaning that need more attention or that have not been explored yet. There are various types of functional dependencies, and as they are widely used to improve data quality, we would like to further investigate the use of dynamic functional dependency discoveries in improving data stream quality.

Despite the existing techniques for cleaning data streams, weak domain knowledge is often left out when cleaning data streams, which may affect the performance of the proposed algorithm and techniques, so we intend to explore how continuous monitoring and real-time feedback can assist in improving the efficiency of the algorithm and techniques. In addition, several techniques have been proposed to clean data streams; nevertheless, cleaning unstructured data streams requires more investigation. In future work, AI-driven data-stream-cleaning techniques can be employed to recognize and validate patterns in the stream data and predict and fill in missing data based on the patterns in the stream, since traditional techniques cannot clean data stream.

Author Contributions: Conceptualization, O.A., E.P. and S.T.; methodology, O.A.; software, O.A.; validation, O.A.; formal analysis, O.A.; investigation, O.A.; data curation, O.A.; writing—original draft preparation, O.A.; writing—review and editing, E.P. and S.T.; visualization, O.A.; supervision, E.P. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Erl, T.; Khattak, W.; Buhler, P. *Big Data Fundamentals: Concepts, Drivers & Techniques*; Prentice Hall Press: Upper Saddle River, UJ, USA, 2016.
2. Kolajo, T.; Daramola, O.; Adebiyi, A. Big data stream analysis: A systematic literature review. *J. Big Data* **2019**, *6*, 47. [CrossRef]
3. Han, J.; Pei, J.; Tong, H. *Data Mining: Concepts and Techniques*; Morgan kaufmann: Burlington, MA, USA, 2022.
4. Ridzuan, F.; Zainon, W.M.N.W. A review on data cleansing methods for big data. *Procedia Comput. Sci.* **2019**, *161*, 731–738. [CrossRef]
5. PRISMA. PRISMA Flow Diagram. Available online: http://www.prisma-statement.org (accessed on 1 July 2023).
6. Turabieh, H.; Mafarja, M.; Mirjalili, S. Dynamic Adaptive Network-Based Fuzzy Inference System (D-ANFIS) for the Imputation of Missing Data for Internet of Medical Things Applications. *IEEE Internet Things J.* **2019**, *6*, 9316–9325. [CrossRef]
7. Sun, D.; Xue, S.; Wu, H.; Wu, J. A Data Stream Cleaning System Using Edge Intelligence for Smart City Industrial Environments. *IEEE Trans. Ind. Inform.* **2022**, *18*, 1165–1174. [CrossRef]
8. Shao, X.; Zhang, M.; Meng, J. Data Stream Clustering and Outlier Detection Algorithm Based on Shared Nearest Neighbor Density. In Proceedings of the 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 25–26 January 2018; pp. 279–282. [CrossRef]
9. Vázquez, F.I.; Zseby, T.; Zimek, A. Outlier Detection Based on Low Density Models. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 970–979. [CrossRef]
10. Yoon, S.; Lee, J.G.; Lee, B.S. NETS: Extremely fast outlier detection from a data stream via set-based processing. *Proc. VLDB Endow.* **2018**, *12*, 1303–1315. [CrossRef]
11. Yuan, G.; Cai, S.; Hao, S. A Novel Weighted Frequent Pattern-Based Outlier Detection Method Applied to Data Stream. In Proceedings of the 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, China, 12–15 April 2019; pp. 503–510. [CrossRef]
12. Alghushairy, O.; Alsini, R.; Ma, X.; Soule, T. A Genetic-based incremental local outlier factor algorithm for efficient data stream processing. In Proceedings of the 2020 the 4th International Conference on Compute and Data Analysis, San Jose, CA, USA, 9–12 March 2020; pp. 38–49. [CrossRef]
13. Alsini, R.; Alghushairy, O.; Ma, X.; Soule, T. A Grid Partition-Based Local Outlier Factor by Reachability Distance for Data Stream Processing. In Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 16–18 December 2020; pp. 369–375. [CrossRef]
14. Gao, J.; Ji, W.; Zhang, L.; Li, A.; Wang, Y.; Zhang, Z. Cube-based incremental outlier detection for streaming computing. *Inf. Sci.* **2020**, *517*, 361–376. [CrossRef]
15. Moon, A.; Zhuo, X.; Zhang, J.; Son, S.W.; Song, Y.J. Anomaly Detection in Edge Nodes using Sparsity Profile. In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data), Atlanta, GA, USA, 10–13 December 2020; pp. 1236–1245. [CrossRef]
16. Yu, Y.; Wu, X.; Yuan, S. Anomaly Detection for Internet of Things Based on Compressed Sensing and Online Extreme Learning Machine Autoencoder. *J. Phys. Conf. Ser.* **2020**, *1544*, 012027. [CrossRef]
17. Zhu, R.; Ji, X.; Yu, D.; Tan, Z.; Zhao, L.; Li, J.; Xia, X. KNN-Based Approximate Outlier Detection Algorithm Over IoT Streaming Data. *IEEE Access* **2020**, *8*, 42749–42759. [CrossRef]
18. Gruhl, C.; Tomforde, S. OHODIN—Online Anomaly Detection for Data Streams. In Proceedings of the 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C), Washington DC, USA, 27 September–1 October 2021; pp. 193–197. [CrossRef]
19. Togbe, M.U.; Chabchoub, Y.; Boly, A.; Barry, M.; Chiky, R.; Bahri, M. Anomalies detection using isolation in concept-drifting data streams. *Computers* **2021**, *10*, 13. [CrossRef]
20. Wang, Q.; Yan, B.; Su, H.; Zheng, H. Anomaly Detection for Time Series Data Stream. In Proceedings of the 2021 IEEE 6th International Conference on Big Data Analytics (ICBDA), Xiamen, China, 5–8 March 2021; pp. 118–122. [CrossRef]
21. Zhao, Z.; Birke, R.; Han, R.; Robu, B.; Bouchenak, S.; Mokhtar, S.B.; Chen, L.Y. Enhancing Robustness of On-Line Learning Models on Highly Noisy Data. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 2177–2192. [CrossRef]

22. Ariyaluran Habeeb, R.A.; Nasaruddin, F.; Gani, A.; Amanullah, M.A.; Abaker Targio Hashem, I.; Ahmed, E.; Imran, M. Clustering-based real-time anomaly detection—A breakthrough in big data technologies. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3647. [CrossRef]
23. Jiang, Y.G.; Kang, C.; Shen, Y.; Huang, T.T.; Zhai, G.D. Research on Argo Data Anomaly Detection Based on Improved DBSCAN Algorithm. In Proceedings of the China Conference on Wireless Sensor Networks, Singapore, 10 November 2022; pp. 44–54. [CrossRef]
24. Benjelloun, F.-Z.; Oussous, A.; Bennani, A.; Belfkih, S.; Ait Lahcen, A. Improving outliers detection in data streams using LiCS and voting. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *33*, 1177–1185. [CrossRef]
25. Xu, X.; Lei, Y.; Li, Z. An Incorrect Data Detection Method for Big Data Cleaning of Machinery Condition Monitoring. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2326–2336. [CrossRef]
26. Najib, F.M.; Ismail, R.M.; Badr, N.L.; Gharib, T.F. Clustering based approach for incomplete data streams processing. *J. Intell. Fuzzy Syst.* **2020**, *38*, 3213–3227. [CrossRef]
27. Shen, L.; He, X.; Liu, M.; Qin, R.; Guo, C.; Meng, X.; Duan, R. A Flexible Ensemble Algorithm for Big Data Cleaning of PMUs. *Front. Energy Res.* **2021**, *9*, 695057. [CrossRef]
28. Lizhen, W.; Yifan, Z.; Gang, W.; Xiaohong, H. A novel short-term load forecasting method based on mini-batch stochastic gradient descent regression model. *Electr. Power Syst. Res.* **2022**, *211*, 108226. [CrossRef]
29. Liu, J.; Cao, Y.; Li, Y.; Guo, Y.; Deng, W. A big data cleaning method based on improved CLOF and Random Forest for distribution network. *CSEE J. Power Energy Syst.* **2020**, 1–10. [CrossRef]
30. Thakur, S.; Dharavath, R. KMDT: A hybrid cluster approach for anomaly detection using big data. In Proceedings of the Information and Decision Sciences: Proceedings of the 6th International Conference on FICTA, Singapore, 14–17 October 2017; pp. 169–176. [CrossRef]
31. Heigl, M.; Anand, K.A.; Urmann, A.; Fiala, D.; Schramm, M.; Hable, R. On the improvement of the isolation forest algorithm for outlier detection with streaming data. *Electronics* **2021**, *10*, 1534. [CrossRef]
32. Rivera, J.J.D.; Khan, T.A.; Akbar, W.; Afaq, M.; Song, W.C. An ML Based Anomaly Detection System in real-time data streams. In Proceedings of the 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2021; pp. 1329–1334. [CrossRef]
33. Degirmenci, A.; Karal, O. Efficient density and cluster based incremental outlier detection in data streams. *Inf. Sci.* **2022**, *607*, 901–920. [CrossRef]
34. Panneerselvam, M.; Neela, K.; Rajeshwari, R.; Vengadapathiraj, M.; Sobitha, S.; Mohanavel, V. A Novel Approach to Identify the Anomaly Detection in Electricity usage based on Machine Learning Algorithms and Big Data. In Proceedings of the 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 20–22 October 2022; pp. 1393–1400. [CrossRef]
35. Prabhakar, T.S.; Veena, M.N. Efficient anomaly detection using deer hunting optimization algorithm via adaptive deep belief neural network in mobile network. *J. Ambient Intell. Humaniz. Comput.* **2022**, 1–17. [CrossRef]
36. Pei, C.; Zhang, S.; Zeng, X. Research on anomaly detection of wireless data acquisition in power system based on spark. *Energy Rep.* **2022**, *8*, 1392–1404. [CrossRef]
37. Xu, B. Power Station Abnormal Data Cleaning Method Based On Big Data Mining. In Proceedings of the 2021 IEEE Sustainable Power and Energy Conference (iSPEC), Nanjing, China, 23 December 2021; pp. 3809–3814.
38. Andreoni Lopez, M.; Mattos, D.M.F.; Duarte, O.C.M.B.; Pujolle, G. A fast unsupervised preprocessing method for network monitoring. *Ann. Des Telecommun./Ann. Telecommun.* **2019**, *74*, 139–155. [CrossRef]
39. Zhang, X.; Lin, R.; Xu, H. An Adaptive Parameters Density Cluster Algorithm for Data Cleaning in Big Data. In Proceedings of the Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, 17–20 July 2020; pp. 543–553. [CrossRef]
40. Fitters, W.; Cuzzocrea, A.; Hassani, M. Enhancing LSTM prediction of vehicle traffic flow data via outlier correlations. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Madrid, Spain, 12–16 July 2021; pp. 210–217. [CrossRef]
41. Arora, S.; Rani, R.; Saxena, N. An efficient approach for detecting anomalous events in real-time weather datasets. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6707. [CrossRef]
42. Iturria, A.; Labaien, J.; Charramendieta, S.; Lojo, A.; Del Ser, J.; Herrera, F. A framework for adapting online prediction algorithms to outlier detection over time series. *Knowl.-Based Syst.* **2022**, *256*, 109823. [CrossRef]
43. Wang, Y.; Perry, M.; Whitlock, D.; Sutherland, J.W. Detecting anomalies in time series data from a manufacturing system using recurrent neural networks. *J. Manuf. Syst.* **2022**, *62*, 823–834. [CrossRef]
44. Zhou, Y.; Xu, K.; He, F.; Zhang, Z. Online abnormal interval detection and classification of industrial time series data based on multi-scale deep learning. *J. Taiwan Inst. Chem. Eng.* **2022**, *138*, 104445. [CrossRef]
45. Albattah, A.; Rassam, M.A. A Correlation-Based Anomaly Detection Model for Wireless Body Area Networks Using Convolutional Long Short-Term Memory Neural Network. *Sensors* **2022**, *22*, 1951. [CrossRef] [PubMed]
46. Belacel, N.; Richard, R.; Xu, Z.M. An LSTM Encoder-Decoder Approach for Unsupervised Online Anomaly Detection in Machine Learning Packages for Streaming Data. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 3348–3357. [CrossRef]

47. Gao, Y.; Yin, X.; He, Z.; Wang, X. A deep learning process anomaly detection approach with representative latent features for low discriminative and insufficient abnormal data. *Comput. Ind. Eng.* **2023**, *176*, 108936. [CrossRef]
48. Smrithy, G.S.; Balakrishnan, R. Automated modeling of real real-time anomaly detection using non -parametric statistical technique for data streams in cloud environments. *J. Commun. Softw. Syst.* **2019**, *15*, 225–232. [CrossRef]
49. Yu, K.; Shi, W.; Santoro, N.; Ma, X. Real-time Outlier Detection Over Streaming Data. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019.
50. Karn, R.; Joshi, S.R.; Bista, U.; Joshi, B.; Baral, D.S.; Shakya, A. Anomaly Detection in Distributed Streams. *Inf. Commun. Technol. Intell. Syst.* **2021**, *196*, 139–147. [CrossRef]
51. Jamshidi, E.J.; Yusup, Y.; Kayode, J.S.; Kamaruddin, M.A. Detecting outliers in a univariate time series dataset using unsupervised combined statistical methods: A case study on surface water temperature. *Ecol. Inform.* **2022**, *69*, 101672. [CrossRef]
52. Kurt, M.N.; Yılmaz, Y.; Wang, X. Sequential Model-Free Anomaly Detection for Big Data Streams. In Proceedings of the 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 24–27 September 2019; pp. 421–425.
53. Bobulski, J.; Kubanek, M. A method of cleaning data from IoT devices in Big data systems. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 6596–6598.
54. Kulanuwat, L.; Chantrapornchai, C.; Maleewong, M.; Wongchaisuwat, P.; Wimala, S.; Sarinnapakorn, K.; Boonya-Aroonnet, S. Anomaly detection using a sliding window technique and data imputation with machine learning for hydrological time series. *Water* **2021**, *13*, 1862. [CrossRef]
55. Fountas, P.; Kolomvatsos, K. A Continuous Data Imputation Mechanism based on Streams Correlation. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–6. [CrossRef]
56. Zhao, X.; Jia, K.; Letcher, B.; Fair, J.; Xie, Y.; Jia, X. VIMTS: Variational-based Imputation for Multi-modal Time Series. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 349–358. [CrossRef]
57. Bimonte, S.; Ren, L.; Koueya, N. A linear programming-based framework for handling missing data in multi-granular data warehouses. *Data Knowl. Eng.* **2020**, *128*, 101832. [CrossRef]
58. Fang, J. Research on automatic cleaning algorithm of multi-dimensional network redundant data based on big data. *Evol. Intell.* **2022**, *15*, 2609–2617. [CrossRef]
59. Jehlol, H.B.; George, L.E. Big Data De-duplication Using Classification Scheme based on Histogram of File Stream. In Proceedings of the 2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE), Hadhramaut, Yemen, 3–5 December 2022; pp. 1–7. [CrossRef]
60. Xiao, B.; Wang, Z.; Liu, Q.; Liu, X. SMK-means: An improved mini batch k-means algorithm based on mapreduce with big data. *Comput. Mater. Contin.* **2018**, *56*, 365–379. [CrossRef]
61. Sun, H.; He, Q.; Liao, K.; Sellis, T.; Guo, L.; Zhang, X.; Shen, J.; Chen, F. Fast Anomaly Detection in Multiple Multi-Dimensional Data Streams. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1218–1223. [CrossRef]
62. Reunanen, N.; Räty, T.; Jokinen, J.J.; Hoyt, T.; Culler, D. Unsupervised online detection and prediction of outliers in streams of sensor data. *Int. J. Data Sci. Anal.* **2020**, *9*, 285–314. [CrossRef]
63. Crépey, S.; Lehdili, N.; Madhar, N.; Thomas, M. Anomaly Detection in Financial Time Series by Principal Component Analysis and Neural Networks. *Algorithms* **2022**, *15*, 385. [CrossRef]
64. Huang, Y.; Du, F.; Chen, J.; Chen, Y.; Wang, Q.; Li, M. Generalized Pareto Model Based on Particle Swarm Optimization for Anomaly Detection. *IEEE Access* **2019**, *7*, 176329–176338. [CrossRef]
65. Surapaneni, R.K.; Nimmagadda, S.; Pragathi, K. Unsupervised Classification Approach for Anomaly Detection in Big Data Streams. *Lect. Notes Netw. Syst.* **2021**, *201*, 71–79. [CrossRef]
66. Zhang, J.; Wang, C.; Li, Z.; Zhang, X. Threshold-free Anomaly Detection for Streaming Time Series through Deep Learning. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 13–16 December 2021; pp. 1783–1789. [CrossRef]
67. García-Gil, D.; Luengo, J.; García, S.; Herrera, F. Enabling Smart Data: Noise filtering in Big Data classification. *Inf. Sci.* **2019**, *479*, 135–152. [CrossRef]
68. Ma, J.; Cheng, J.C.P.; Jiang, F.; Chen, W.; Wang, M.; Zhai, C. A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data. *Energy Build.* **2020**, *216*, 109941. [CrossRef]
69. Li, G.; Wang, J.; Liang, J.; Yue, C. The application of a double CUSUM algorithm in industrial data stream anomaly detection. *Symmetry* **2018**, *10*, 264. [CrossRef]
70. Rollo, F.; Bachechi, C.; Po, L. Semi Real-time Data Cleaning of Spatially Correlated Data in Traffic Sensor Networks. In Proceedings of the 18th International Conference on Web Information Systems and Technologies-WEBIST, Valetta, Malta, 25–27 October 2022; pp. 83–94.
71. Zhu, Y.; Xie, C. Edge-Cloud Hybrid Tiny Data Reduction Model for Anomaly Detection. In Proceedings of the 2022 IEEE International Conference on e-Business Engineering (ICEBE), Bournemouth, UK, 14–16 October 2022; pp. 51–57. [CrossRef]

72. Yang, C.; Du, Z.; Meng, X.; Zhang, X.; Hao, X.; Bader, D.A. Anomaly Detection in Catalog Streams. *IEEE Trans. Big Data* **2023**, *9*, 294–311. [CrossRef]
73. Amen, B.; Grigoris, A. Collective Anomaly Detection Using Big Data Distributed Stream Analytics. In Proceedings of the 2018 14th International Conference on Semantics, Knowledge and Grids (SKG), Guangzhou, China, 12–14 September 2018; pp. 188–195. [CrossRef]
74. Chen, Z.; Yu, X.; Ling, Y.; Song, B.; Quan, W.; Hu, X.; Yan, E. Correlated Anomaly Detection from Large Streaming Data. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 982–992. [CrossRef]
75. Manjunatha, H.C.; Mohanasundaram, R. BRNADS: Big data real-time node anomaly detection in social networks. In Proceedings of the 2018 2nd International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 19–20 January 2018; pp. 929–932. [CrossRef]
76. Su, S.; Xiao, L.; Ruan, L.; Xu, R.; Li, S.; Wang, Z.; He, Q.; Li, W. ADCMO: An Anomaly Detection Approach Based on Local Outlier Factor for Continuously Monitored Object. In Proceedings of the 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), Xiamen, China, 16–18 December 2019; pp. 865–874. [CrossRef]
77. Cao, K.; Liu, Y.; Meng, G.; Liu, H.; Miao, A.; Xu, J. Trajectory Outlier Detection on Trajectory Data Streams. *IEEE Access* **2020**, *8*, 34187–34196. [CrossRef]
78. Dias, R.; Mauricio, L.A.F.; Poggi, M. Toward an Efficient Real-Time Anomaly Detection System for Cloud Datacenters. In Proceedings of the 2020 IFIP Networking Conference (Networking), Paris, France, 22–26 June 2020; pp. 529–533.
79. Borah, A.; Gruenwald, L.; Leal, E.; Panjei, E. A GPU Algorithm for Detecting Contextual Outliers in Multiple Concurrent Data Streams. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; pp. 2737–2742. [CrossRef]
80. Dani, Y.; Gunawan, A.Y.; Indratno, S.W. Detecting Online Outlier for Data Streams using Recursive Residual. In Proceedings of the 2022 Seventh International Conference on Informatics and Computing (ICIC), Denpasar, Bali, Indonesia, 8–9 December 2022. [CrossRef]
81. Leigh, C.; Alsibai, O.; Hyndman, R.J.; Kandanaarachchi, S.; King, O.C.; McGree, J.M.; Neelamraju, C.; Strauss, J.; Talagala, P.D.; Turner, R.D.R.; et al. A framework for automated anomaly detection in high frequency water-quality data from in situ sensors. *Sci. Total Environ.* **2019**, *664*, 885–898. [CrossRef] [PubMed]
82. Souza, T.I.A.; Aquino, A.L.L.; Gomes, D.G. A method to detect data outliers from smart urban spaces via tensor analysis. *Future Gener. Comput. Syst.* **2019**, *92*, 290–301. [CrossRef]
83. Gupta, G.P.; Khedwal, J. Framework for Error Detection & its Localization in Sensor Data Stream for reliable big sensor data analytics using Apache Spark Streaming. *Procedia Comput. Sci.* **2020**, *167*, 2337–2342. [CrossRef]
84. Zheng, H.; Tian, B.; Liu, X.; Zhang, W.; Liu, S.; Wang, C. Data Quality Identification Model for Power Big Data. In Proceedings of the International Conference of Pioneering Computer Scientists, Engineers and Educators, Singapore, 10 August 2022; pp. 20–29. [CrossRef]
85. Wang, T.; Ke, H.; Zheng, X.; Wang, K.; Sangaiah, A.K.; Liu, A. Big Data Cleaning Based on Mobile Edge Computing in Industrial Sensor-Cloud. *IEEE Trans. Ind. Inform.* **2020**, *16*, 1321–1329. [CrossRef]
86. You, D.; Wu, X.; Shen, L.; Chen, Z.; Ma, C.; Deng, S. Online Feature Selection for Streaming Features with High Redundancy Using Sliding-Window Sampling. In Proceedings of the 2018 IEEE International Conference on Big Knowledge (ICBK), Hefei, China, 9–10 August 2017; pp. 205–212. [CrossRef]
87. Pezoulas, V.C.; Kourou, K.D.; Kalatzis, F.; Exarchos, T.P.; Venetsanopoulou, A.; Zampeli, E.; Gandolfo, S.; Skopouli, F.; De Vita, S.; Tzioufas, A.G.; et al. Medical data quality assessment: On the development of an automated framework for medical data curation. *Comput. Biol. Med.* **2019**, *107*, 270–283. [CrossRef]
88. Salloum, S.; Huang, J.Z.; He, Y. Exploring and cleaning big data with random sample data blocks. *J. Big Data* **2019**, *6*, 45. [CrossRef]
89. Ju, X.; Lian, F.; Zhang, Y. Data Cleaning Optimization for Grain Big Data Processing using Task Merging. In Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 20–22 December 2019; pp. 225–233.
90. Ding, X.; Qin, S. Iteratively modeling based cleansing interactively samples of big data. In Proceedings of the Cloud Computing and Security: 4th International Conference, ICCCS 2018, Haikou, China, 8–10 June 2018; pp. 601–612. [CrossRef]
91. Rama Satish, K.V.; Kavya, N.P. Hybrid optimization in big data: Error detection and data repairing by big data cleaning using CSO-GSA. In Proceedings of the International Conference on Cognitive Computing and Information Processing, Bengaluru, India, 15–16 December 2017; Springer: Berlin, Germany, 2018; Volume 801, pp. 258–273. [CrossRef]