electronics

MDPI

Review

Outlier Detection in Streaming Data for Telecommunications and Industrial Applications: A Survey

Roland N. Mfondoum, Antoni Ivanov *, Pavlina Koleva, Vladimir Poulkov and Agata Manolova

Faculty of Telecommunications, Technical University of Sofia, bul. Kl. Ohridski 8, 1000 Sofia, Bulgaria; mfondoum2000@gmail.com (R.N.M.); p_koleva@tu-sofia.bg (P.K.); vkp@tu-sofia.bg (V.P.); amanolova@tu-sofia.bg (A.M.)

* Correspondence: astivanov@tu-sofia.bg

Abstract: Streaming data are present all around us. From traditional radio systems streaming audio to today's connected end-user devices constantly sending information or accessing services, data are flowing constantly between nodes across various networks. The demand for appropriate outlier detection (OD) methods in the fields of fault detection, special events detection, and malicious activities detection and prevention is not only persistent over time but increasing, especially with the recent developments in Telecommunication systems such as Fifth Generation (5G) networks facilitating the expansion of the Internet of Things (IoT). The process of selecting a computationally efficient OD method, adapted for a specific field and accounting for the existence of empirical data, or lack thereof, is non-trivial. This paper presents a thorough survey of OD methods, categorized by the applications they are implemented in, the basic assumptions that they use according to the characteristics of the streaming data, and a summary of the emerging challenges, such as the evolving structure and nature of the data and their dimensionality and temporality. A categorization of commonly used datasets in the context of streaming data is produced to aid data source identification for researchers in this field. Based on this, guidelines for OD method selection are defined, which consider flexibility and sample size requirements and facilitate the design of such algorithms in Telecommunications and other industries.

Keywords: outlier detection; streaming data; 5G; IoT; computational complexity; machine learning; deep learning

Citation: Mfondoum, R.N.; Ivanov, A.; Koleva, P.; Poulkov, V.; Manolova, A. Outlier Detection in Streaming Data for Telecommunications and Industrial Applications: A Survey. *Electronics* **2024**, *13*, 3339. https://doi.org/10.3390/electronics13163339

Academic Editors: Agnieszka Konys and Agnieszka Nowak-Brzezińska

Received: 24 June 2024
Revised: 13 August 2024
Accepted: 17 August 2024
Published: 22 August 2024

1. Introduction

In the real world, various technologies and applications generate large volumes of timestamped data at high speed. These streams of information, also known as streaming data, are generated by industrial processes [1], Internet of Things (IoT) devices in the sports and health industries, sensors monitoring the environment [2], crops in the agricultural sector [3], autonomous systems sensors such as self-driving cars in the transport industry, telecommunication networks and sensors [4], payments and trading systems in the financial sector [5,6], and many more fields. Outlier detection (OD) methods consider the sudden changes, anomalies, rare occurrences, and exceptions in radio frequency (RF) samples or user behavior data and use them to improve the resource allocation or system control functions of wireless communication networks in real-time [7]. Such applications include functions related to resource management, orchestration and customization in wireless networks, for example, covariance-based detection of inaccurate RF measurements in spectrum sensing and countering the falsification of local measurements by malicious nodes in cognitive radio (CR) networks, histogram-based analysis of the location-based measurement accuracy of Wireless Sensor Networks (WSNs), deep learning (DL)-based key performance indicator (KPI) prediction for user profile classification, clustering of user data records for anomalous events prediction in mobile networks [8], traffic classification in IoT networks [9], mean likelihood-based prediction of process-specific data in

industrial IoT mobile networks, and machine learning (ML)-based traffic demand and user behavior prediction for proactive radio resource management in fifth-generation (5G) mobile networks.

The goal of this paper is to conduct a systematic review of the literature on OD in streaming data in general, with a specific focus on applications in the information technology (IT) and telecommunications industries. Additionally, it provides researchers with an overview of the detection methods across papers, as well as their key assumptions, and links them to the type of outlier they are best suited for based on their assumptions and their application. Finally, it concludes with a mapping of the advantages and disadvantages of various methods and the list of publicly available datasets used in the literature for training, testing, and performance benchmarking.

1.1. Literature Review Strategy

The systematic literature review strategy described in Figure 1 was used for this article on OD in streaming data with a specific focus on applications in the telecommunications sector and other industries. While searching for the relevant literature, specialized databases such as Google Scholar, Institute of the Electrical and Electronic Engineers (IEEE) journals, and Science Direct were used. The keywords "outlier detection", "anomaly detection", "streaming data", "computational efficiency", "high-dimensionality", "concepts", and "methods" were employed in the initial search. Then, surveys or literature review papers were filtered to those published within the last 5 years, while papers related to applications, state-of-the-art methods, and improvement were only filtered with regards to their relevance while the year of publication was not considered. The output was then reviewed by reading the introduction, contributions, and conclusions of the papers to identify which ones were subject to the relevant criteria of streaming data, computational efficiency, and high-dimensional data, or applications in telecommunications and other industries. With this approach, the relevant papers were selected and then read to extract information on the methods used, the areas of application, and the challenges described.

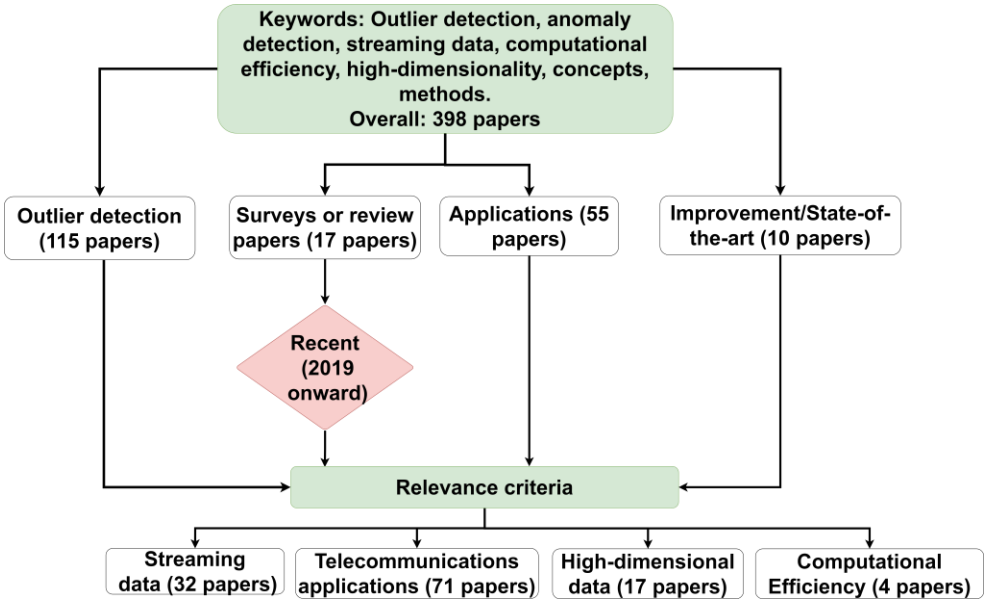

Figure 1. Literature review strategy.

1.2. Structure of This Paper

The rest of this paper is organized as indicated in Figure 2. Section 2 describes the characteristics of streaming data and outliers and introduces the problem of their detection, as well as how it improves in related surveys, and summarizes the contributions of this work. Then, Section 3 surveys the broad categories of OD, including statistics-based OD,

ML-based OD, and DL-based OD, i.e., methods that include the training of neural networks (NNs). The applications of the OD methods are summarized in Section 4. Consequently, Sections 5 and 6 describe the types of outliers and the assumptions that are made for the OD, as well as the performance metrics, respectively. Section 7 provides the advantages and disadvantages of OD. The key findings of this survey and their implications for future work in OD algorithm design and development are summarized in Section 8. This paper is concluded in Section 9.

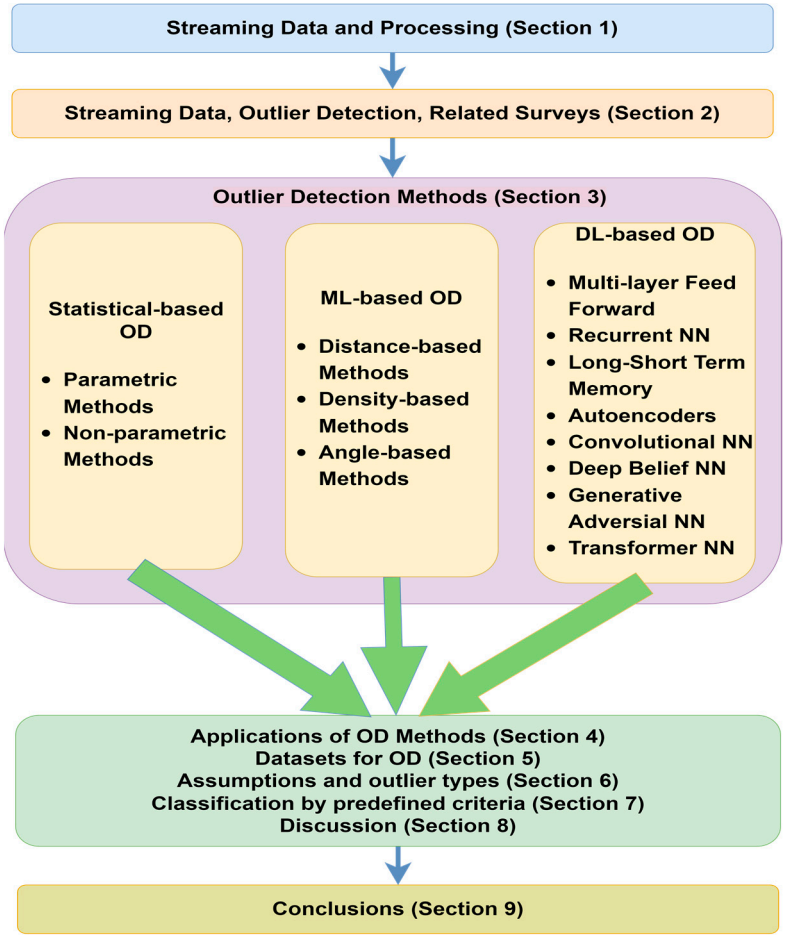

Figure 2. Structure of this paper.

1.3. Contributions

This paper surveys OD in streaming data in a holistic manner, accounting for all the limitations encountered in various related surveys, and facilitates the future development of OD methods by providing the following contributions:

- A summary of the key challenges related to OD in streaming data with a focus on data processing approaches, computational complexity, and the optimal types of detectable outliers. This will facilitate the choice of the best method for specific detection problems based on a particular set of assumptions.
- An overview of the advantages and disadvantages of OD methods for streaming data. The methods are organized into broader categories (statistics, ML, and DL) to facilitate the selection of the most appropriate method depending on the detection challenges, i.e., the available data, sample size requirements, the method's capability to process the temporal changes in the data, and its computational complexity.
- A summary of the applications of OD methods in the telecommunications industry and other sectors is given, as well as datasets commonly used for training, testing, and

benchmarking in the literature. This will aid researchers in identifying the right data sources for developing and testing new OD methods in different fields.

2. Basics of Streaming Data, Outlier Detection, and Related Surveys

2.1. Streaming Data and Processing

Streaming data differ from data at rest by the fact that they flow continuously, without defined boundaries, and can change over time. In essence, streaming data are unbound and dynamic. While the unbound characteristic is straightforward and easy to understand, it is important to clarify what a dynamic data stream means. Streaming data are not always predictable, and the number of features (or classes) within the data may change over time. For instance, a new feature may appear in the data, or an existing feature may vanish. Classes within an existing feature might increase or decrease over time. These appearances and disappearances of classes are known as concept evolution [10]. Streaming data distributions can also change over time, and this is known as concept drift [11]. Both concept drift and concept evolution describe the dynamic characteristic of streaming data. This particularity of streaming data poses challenges to the use of data processing and analytic methods, designed around finite data at rest (i.e., the complete dataset stored securely), without accounting for streaming data variability. Additionally, the flow rate of streaming data has been both increasing and varied due to continuously expanding data volume and velocity (number of events per second), influenced by advancements in data storage and transmission throughput in the IT and telecommunications industries, as well as the variable data production rate by various sources. Moreover, the unbound nature of streaming data poses computational challenges to computer systems, which are naturally limited in storage space, buffering capacity, and processing power. With all these considerations in mind and OD already being complex enough with static data, streaming data pose even more challenges. It is, therefore, imperative to extend existing OD methods to streaming data or develop new detection methods and data management approaches. Such methods should consider the nature of streaming data, which is dynamic as opposed to data at rest and therefore affects the data processing mechanism used in the context of streaming data. The common approach for data pre-processing is to split the data stream into windows and apply existing or adapted detection methods to each of them. Questions related to the determination of the window size and boundary have led to the definition of several windowing mechanisms:

- Time-based windowing—timeslots to delimit windows and data that arrive within a particular timeframe is then considered as a complete subset of the data stream. This subset is processed through analytics and aggregations performed within its boundaries. Given any data stream generating data over time t, the stream can be divided into time windows of time length c starting from the time t_0. The window size c is generally fixed and user-defined in most of the literature based on empirical data, but it can also be dynamic via a systematic approach. This mechanism allows precise control over time intervals and is suitable for time-sensitive data, but it is not adaptable to variable data rates and requires time synchronization [12–14].
- Count-based windowing—this mechanism is based on the number of observations received rather than the duration of the windows. The number of observations n is set as the count threshold based on empirical data and a window is considered completed every time the number of observations since the last window reaches the user-defined threshold [15,16]. This approach is suitable for a data stream with a known pattern and from which analysis is sensitive to the sample size. This mechanism is simple to implement and is effective for streams with constant event rates but is inflexible in handling time variability. It is also unsuitable for time-sensitive analysis. It can be used for detecting peaks or for frequency estimation in a data stream.
- Landmark-based windowing—In this case, data drive the way the window is selected. This method is used, for instance, in the field of network data analysis for security [17], where streams are segmented by flow and analyzed for intrusion detection. Traffic

flows are identified within communication sessions and used as complete sliding windows. The advantages of this mechanism are that it allows for flexible start points and is suitable for event-triggered analysis. However, it has increased complexity in managing dynamic window boundaries and its window size is potentially unbounded.

The following window types for application in the windowing mechanisms presented above can be found in the literature:

- A sliding window [18–20] is used to delete all data and refresh the window slide with the most recent data, for which window size is fixed, assuming that all slides have equal importance. It can be time-based or count-based.
    - A special type of count-based sliding window is the eviction window for which a fixed number of data points overlap between the slides and observations are refreshed based on an eviction policy.
- A damped window [21–23] is based on the assumption that recent data are more important than older data and operates by assigning a higher weight to the most recent data [24].
- A landmark window [23], also known as the time fading window approach, considers data between a fixed point in time in the future called landmark and the current time.
    - Session windows are special types of landmarks for which the boundaries of the window are defined by the session start and end points.
- A tumbling window, for which the slide distance d is equal to its size s, indicative of non-overlapping consecutive windows, $s = d$. This presents the advantage of performing a small cross-sectional analysis on each window.
- Hopping using a fixed window size and a fixed-length step between windows. When the jump length is shorter than the window size, an overlap is maintained between windows; otherwise, there will be gaps between windows and potentially missed data points.

Table 1 presents a comparison of the window types according to the mechanisms they are used in and the way in which their boundaries are defined. In this Table, streaming data window types are grouped by mechanism and classified by the following five criteria:

- Boundary: indicating if the windows' starting and ending points are fixed or variable.
- Overlap: indicating if a data overlap is observed while moving from one window to another.
- Number of passes: indicating if an observation is likely to be maintained across multiple windows.
- Sequence tracking: indicating if the window mechanism requires keeping track of the window sequence.
- Historical data tracking: indicative of whether the window type needs to keep track of information on previous windows such as session identity number.

The Table shows that except for the tumbling windows, all other mechanisms allow overlap between windows, making them memory-intensive. Additionally, any computation performed will be executed more than once on the data (number of passes) when windows overlap, leading to higher computational complexity. Moreover, to implement a particular window mechanism, it is required to keep track of either the time or the data points of the previous window (i.e., the historical data) to define the following window, which adds to the computational complexity when processing streaming data. When it comes to the landmark-type window, it is theoretically undetermined regardless of whether there is an overlap or historical data tracking, and regardless of the number of passes. This is because the landmark point is determined on a case-by-case basis unless it is applied for the whole session (i.e., the landmark is defined based on the activity performed in the specific application), as described above. Consequently, this is the only window type with dynamic boundaries. Therefore, it may be used for applications that require lower computational complexity, while time tracking is still available.

Table 1. Window type comparison according to the mechanisms they are used in and the way in which their boundaries are defined.

Type	Mechanism	Boundaries	Overlap	Number of Passes	Sequence Tracking	Historical Data Tracking
Tumbling	Time-based	Fixed	No	One	No	No
Damped	Time-based	Fixed	Yes	More than one	Yes	Yes
Hopping	Time-based	Fixed	Yes	More than one	No	Yes
Sliding	Time-based	Fixed	Yes	More than one	Yes	No
Sliding (Eviction)	Count-based	Fixed	Yes	More than one	No	Yes
Landmark	Landmark-based	Dynamic	Undetermined	Undetermined	Yes	Undetermined
Landmark (Session)	Activity-based	Dynamic	Yes	More than one	No	Yes

Throughout this paper, computational complexity will be expressed using the notation O, which will generally be represented as a function of the dataset size n and dimension k, or other parameters to be defined when needed.

2.2. Data Dimensionality

Dimensionality is an additional component of streaming data that may complexify its processing. Some papers classify data as low and high-dimensional spaces depending on the number of features or dimensions d versus the number of observations n. The size of a dataset corresponds to the tuple $(n \times d)$, representing the size of the dataset. A high-dimensional dataset is defined as one for which the number of features d is greater or equal to the number of observations n, $d \geq n$ [25]. Some papers consider a dataset as high-dimensional if it contains 10 or more features [26]. High-dimensional data are often mentioned in the literature dealing with imaging or pattern analysis such as in the field of mass spectrometry [27], geospatial imagery analysis [28], and more. Given the small amount of data used, a challenge of high-dimensional data is the reproducibility [29] of the results. Through this paper, we will classify any dataset with 1–4 features as low-dimensional, 5–9 as medium-dimensional, and any with 10 or more features as high-dimensional. As indicated in the literature, low-dimensional datasets comprise fewer than 10 dimensions [30]. In some cases, especially those related to distance-based and most statistical-based methods, good performance has been reported for dimensionality $d \leq 5$. A further explanation for this classification is given in Section 7.1.

2.3. Outlier Characteristics and Detection

An outlier is commonly defined as an observation that deviates greatly from the rest of the data distribution, leading to the hypothesis that it does not belong to it [31]. Three different types of outliers are mentioned in the literature: global, contextual, and collective outliers [32–34]. An outlier can be of one or more types [35] depending on the dataset. Global outliers, also known as point outliers, are detected by analyzing the entire data space to identify observations that deviate from the normal distribution of the data. These outliers represent one or more points that are considered anomalous within the context of the entire dataset, considering only its spatial distribution. Local outliers represent one or more static outliers within a data subset with only spatial awareness. Contextual outliers use both the context of the observation (e.g., time, date, and location) and its behavior (e.g., the actual information measured) to determine their outlierness. In this case, the spatiotemporal context plays a role in discriminating data that otherwise would appear normal using the global detection approach. This category represents one or more dynamic outliers within a subset determined by a specific time or space context. A further group of contextual outliers are the point outliers in temporal data and subsequent outliers [36].

Collective outliers try to capture the concepts of frequency of occurrence and group outliers from within a dataset. A group of outlying observations with spatial proximity [37], which are also spread across a few time intervals [38] within the same dataset, are considered collective outliers. This type includes more than one (both static and dynamic) outlier generated by the same process and is an effect of an underlying problem that generates abnormal observations. They are observed in systematic human errors or machinery faults in industrial processes, which generate erroneous values.

OD is the process of identifying deviation from within a dataset produced by the same data generation mechanism. Any random error or data generated by an interfering process different from the observed data can be classified as noise. Noisy data carry no meaningful information in the context of analysis and should be filtered out of the observed data before performing any analysis, while outliers need to be dealt with more cautiously depending on the analytic requirements and may not always be removed. Furthermore, prior knowledge [39] of the data's behavior could be an asset in determining if new observations are consistent with previously observed data or if they are candidate outliers. This prior knowledge of the data, also known as "a priori", is used as the ground truth, which in temporal data constitutes frequent patterns [40] versus surprising patterns [41,42] represented by outliers over time. Several references are related to anomaly detection and this term is sometimes used interchangeably with OD. However, it is important to make a slight differentiation between the two terms. Although the process for detecting anomalies is similar to that for OD [43,44], not all outliers are anomalies. The key difference is that some outliers are simply rare but valid observations, while anomalies typically indicate invalid, erroneous, or failure-related data. Consequently, the mechanisms for handling them differ. Anomalies are generally suppressed or flagged for immediate attention, whereas outliers may be addressed through different approaches after detection. Since this paper focuses solely on the detection phase and does not address post-detection handling, "anomaly detection" should be understood to refer to OD.

2.4. Related Surveys

This paper presents a survey of OD in streaming data using the most common statistical, ML, and DL methods found in the literature, including their major applications in various industries, their fitness for the OD task based on various characteristics such as efficiency, computational performance, data type, and data dimensionality, and the type of outlier they are most likely to detect. Several preceding OD surveys are summarized here (Table 2) in terms of their scope, advantages, and limitations.

Table 2. Comparison of relevant surveys on OD methods and their applications.

Survey	Scope of Reference Survey	Advantages	Limitations	Our Contributions
Duraj, A., et. al. [45]	Experimental study comparing the performance of a few statistical algorithms for OD.	Conducted a detailed performance analysis allowing the comparison of methods by OD count and algorithm time.	Analysis is limited to univariate data analysis.	Our survey covers *OD in multivariate data.*
Fernandes, G., et al. [46]	A comprehensive survey on network anomaly detection.	Covers 5 domains of network anomaly detection.	Applications are limited to the security perspective and do not account for events or fault detection.	Our survey provides a broader coverage of detection by *outlier types and applications on various types of outliers by industry.*

Table 2. *Cont.*

Survey	Scope of Reference Survey	Advantages	Limitations	Our Contributions
Dwivedi, R.K., et. al. [47]	Outlier detection strategies in WSNs.	Covers most OD methods in detail and proposes a structured classification of outlier source by noise/error, malicious attack, or specific events.	Does not provide guidance on OD method selection by problem type.	Our survey provides *OD method selection guidelines*, based on streaming data characteristics and detection tasks.
Dwivedi, R.K., et. al. [47]	Survey of Machine learning-based OD methods in WSNs.	Detailed review of ML methods and their application for OD in WSNs.	Limited to ML Methods	Our survey covers a broader range of methods including *statistical-*, *ML-*, and *DL*-based methods.
Wang, H., et. al. [48]	Survey of OD methods proposed between 2000–2016.	Provides an extensive analysis of most methods, tools, datasets, and performance metrics used in the literature.	Slightly covers DL methods and does not cover applications.	Our survey provides a summary of the applications and covers recent DL methods including *state-of-the-art transformers*.
Habeeb, R. A. A., et al. [49]	Survey of real-time anomaly detection in Big Data.	Provides a detailed taxonomy for classifying studies performing anomaly detection in Big Data by methods used, applications, technology, and type of outliers.	Does not cover the computational complexity of the method in detail. Does not cover the advantages and disadvantages of methods by streaming data challenge.	Our survey categorizes the *computational complexity* by method.
Samara, M.A., et. al. [50]	Survey of OD in IoT networks	Provides a summary of challenges of OD in IoT.	Applications are limited to IoT networks.	Our survey covers a wider range of *Telecommunications and industrial applications*.
Gaddam, A., et. al. [51]	Fault Detection in IoT networks.	Provides a methodology for sensor faults determination in IoT using various methods.	Applications are limited to IoT fault detection. Does not cover concept evolution cases.	Our survey outlines OD methods for *handling concept drift or evolution* in data streams.
Souiden, I., et. al. [52]	OD in high-dimensional data.	Developed 2 taxonomies of OD methods in high-dimensional data.	Does not provide applications per OD method or recommendations on method selection.	Our survey categorizes *OD methods by application* and provides several *OD selection recommendations based on the data structure*.

Even though these surveys provide a helpful overview of OD in select fields, none of them combine the categorization of OD methods, key assumptions in relation to detection according to streaming data challenges, advantages, and disadvantages, computational complexity, evaluation metrics, and review of their use by application. These gaps are addressed in this paper as indicated in the Contributions column of Table 2.

3. Methods for OD

3.1. Statistical-Based Methods

The statistical-based OD (SBOD) approach is one of the traditional OD methods and assumes that every dataset has an underlying distribution and the probability of each observation of belonging or not to the distribution can be computed and tested. There are two major approaches to SBOD, namely the parametric and non-parametric methods.

Parametric methods are grouped according to their applicability to data by their dimensionality. We distinguish here between univariate and multi-variate data. A summary of the most important parametric methods is given as follows:

- Gaussian-based OD (GBOD) is one of the most frequently used parametric methods that assumes normally distributed data around its mean with a variance σ^2, denoted as $N(,\sigma^2)$. Among the GBOD applied to univariate distributions, the Z-score method gives a measure distance to the mean of each observation by standard deviation [53]. The standard deviation (SD) method is similar to Z-score, but it is the raw untransformed data that are used. Further, the extreme studentized deviate (ESD) method, also called Grubbs' test [54], assumes near normality and removes a user-defined number of outliers n from a data set X by conducting a series of tests T that identify each point with the maximum deviation from the mean while removing them iteratively from the data, i.e., $T_i = \frac{\max\{|x-\bar{x}_i|\}}{\sigma_i}$, where $x \in X,\ i = 1, \ldots n$.
- These methods apply to a univariate distribution centered around their mean and for which normal data lie within a standard deviation of k from the mean, i.e., $\pm k\sigma$. Gaussian methods are more suitable for global OD and are sensitive to extreme values as they use the mean and the standard deviation [55] in their detection process. They are not designed for temporal/sequential dependency and perform best on univariate numerical data with large sample sizes [56]. Popular variants of ESD are generalized ESD (GESD), online sequential ESD [54], and seasonal hybrid ESD (SHESD) [57,58], which are robust for data with a high percentage of outliers. Two major drawbacks of ESD are that the outlier threshold is user-defined, and that its iterative detection approach will make it too slow for time-sensitive detection in streaming data.
- The Boxplot-based (BPOD) method, also known as Tukey's method, was proposed in 1977 [59] and leverages the Boxplot to detect extreme values at both whiskers [60–62]. It uses the median as the central tendency measure and the inter-quartile range (IQR) to measure the dispersion of the data and is more suitable for univariate symmetric distributions [62].
- The Median Absolute Deviation (MAD) is another method based on the median that uses, as a scale estimator, the overall median M_e of all observations' deviation to the median of the distribution, as follows: $M_e = median \vee x_i - median(x) \vee$, i = 1, 2, …, n, where x_i is the observation number i. A point will be classified as an outlier if it is $c * M_e$ distance away from the distribution's median, denoted as $dian(x) \pm c * M_e$, with c higher than 2, 2.5, or 3. MAD assumes that the dataset has a symmetric distribution [63]. Both BPOD and MAD methods are based on robust statistics, assume a symmetric distribution of the data, and are less sensitive to outliers when compared to the Z-score and the SD-based methods. Their efficiency on a Gaussian distribution, also known as Gaussian efficiency [63], are 37% for MAD and up to 82% for BPOD based on the IQR. This indicates that unless the distribution is preliminary known to be skewed, its skewness is to be validated.
- Regression-based OD (RBOD) is another commonly used parametric method that fits a regression model to the data, estimates the residuals, and flags observations with larger residuals as outliers [64]. This method is suitable for OD on multivariate data streams within the sub-space of a sliding window, and it detects outliers in the context of a dependency between univariates while completely ignoring uncorrelated dimensions. Additionally, in real-world data, regression assumptions, i.e., the linearity of the relationship between univariates, the independence and non-multicollinearity of univariates, and the normality and homoscedasticity of the residuals, generally do not hold, making it challenging to use data with no prior information on the distribution. RBOD is suitable for contextual OD and can be used for global OD if all dimensions are considered in the regression model.
- Copula-based OD (COPOD) is another method based on the concept of copulas first defined by Sklar (1959) [65] that focuses on capturing the correlation between univariates of multivariate distributions with continuous marginals. The copula theorem stipu-

lates in simple terms that for any multivariate random distribution, a joint distribution function can be derived as a combination of the hidden link between the marginal distributions, called copula, and the actual marginal distributions [66]. This separation allows analysis of the correlation between the distributions without a priori information about the actual distribution of the marginals. Another important component of this theorem is that if the random variables are continuous, then the copula is unique, which means this can be used to uniquely represent the cumulative distribution of multivariate random variables in the interval $[0, 1]$. The major families of copulas are independence copulas, applicable when the marginals are independent of each other, elliptical copulas based on standard distributions such as the Gaussian or Student-T, and Archimedean copulas, popular in representing multivariate distributions in the real world for their ability to model lower tail dependency (Clayton copula), asymmetric tail dependency (Gumbel copula), or symmetric with both negative and positive dependencies (Frank copula). These copulas use one or more parameters to express the correlation between marginals. A nonparametric family of copulas, known as empirical copulas, are used in studies for OD [67,68], where no assumption on the dependency function is made but instead the data are used to construct such a function. An advantage of using this method is that it may be used without making any prior assumption of the marginal distributions [69] and is applicable to the analysis of heavy-tailed non-linear dependencies [70]. Their applications related to OD have been explored in the field of dimensionality reduction, synthetic data generation, or signal denoising by baselining and decoupling [71,72]. A drawback of the copula is that it represents only the dependence between the marginals and infers no information about them. Additionally, selecting the appropriate copula distribution to use for data representation is non-trivial, hence [69] proposes an unsupervised copula selection algorithm for OD. COPOD is suitable for contextual outliers as the outlierness is determined in the context of the inherent link between the univariates.

- The Gaussian Mixture Model (GMM) is a weighted multivariate Gaussian model, which assumes that any multivariate distribution is a weighted combination of its univariate marginal distributions that all follow a Gaussian distribution [73]. This model relies on Expectation Maximization (EM), which is an iterative algorithm that first estimates the log likelihood of any data point belonging to a prior distribution [74] and then uses the maximum likelihood estimator to find the optimum log likelihood [75] under the assumption that the posterior distribution is known. Major challenges of the GMM models in detecting outliers in streaming data are reflected in their time greediness and requirement for multiple passes on the data to estimate the model. Additionally, the model requires user-defined parameters for its initialization as well as for the training phase. It is therefore suitable for multidimensional numeric data streams comprising historical data, from which the GMM parameters can be initialized and a prior distribution constructed. In this manner, a distribution can be learned from a live data stream before applying the OD on new data once the model is initialized. Since GMM-OD detects outliers by constructing a joint distribution and estimating the likelihood of belonging of each point to this distribution, it is very suitable for OD in multimodal distributions. These outliers are then considered as contextual because their outlierness is subject to the degree to which they are associated with the mixture model. It can be used for global OD as well.

Non-parametric methods are developed to overcome the limitations of the parametric approach when the distribution is not normal (skewed) or when the data are nominal or ordinal [76]. These methods are also known as distribution-free and are sometimes the only options when the normal assumption of parametric methods does not hold. Several non-parametric methods are used for detecting outliers in data. The most common ones are the histogram-based and kernel-based methods.

- Histogram-based OD (HBOD) relies on the frequency of a continuous data distribution to compute a histogram of each continuous feature or a relative histogram of the

frequency of all categories for each categorical feature in any multivariate distribution [77]. The approach for building a histogram h of a feature f of a dataset with n observations starts with values ordering and the definition of the number of bins k that will be used to build the histogram. Then, the data are grouped into the same number of ordered observations $\frac{n}{k}$, which are then represented. The shape of the distribution is impacted by the histogram bin width selection, which determines the density of the histogram, i.e., the bins' width is inversely proportionate to the density. The bin width can either be fixed or dynamic and the latter is recommended [75] for real-time data if the distribution is unknown and there are peaks or periods without data, which would result in gaps in the histogram if a fixed bin is used. The author points out that dynamic bin selection is less sensitive to such extremes or to outliers.

- The kernel-based OD (KBOD) method was introduced in 2007 [78] and is based on the identification of the density of points around each point of a dataset in an Euclidean space using the kernel density estimator (KDE) as the nonparametric density function and as the basis to mark an observation as an outlier. KBOD assumes non-negative, symmetrical, and normally distributed data. The kernel density is used in combination with the reachability distance estimator and the local density estimates (LDEs) of all neighbors of an observation, from which the local density factor (LDF), a continuous measurement of the risk of outlierness, is calculated. The authors of [78] report greater performance of KBOD at detecting local outliers over other methods such as the local outlier factor (LOF) and local correlation integral (LOCI). One of the challenges of KBOD is the impossibility of detecting outliers in multimodal distributions, as a point considered normal in one of the dimensions would be marked as an outlier in other sub-distributions of the dataset. KDE relies on the nearest neighbors [79] to determine a point density as opposed to the LOF which uses the full space. This makes KBOD a good candidate for contextual outliers, even though it can detect global outliers as well.

3.2. ML-Based Methods

A summary of the most prominent ML-based OD methods is given as follows:

- The distance-based OD (DBOD) method is a supervised ML method that leverages the spatial distribution of points in the space S and assumes that in an n-dimensional dataset, a distance d can be calculated between the observations. It uses distance metrics such as the Euclidean, Mahalanobis, and the cosine similarity or the Manhattan distance. The basic assumption of proximity-based methods is that similarity is defined by spatial closeness and the higher the dissimilarity measure or distance, the more likely the observation is an outlier. Any point in the space S is therefore considered normal if it has at least k neighbours at a maximum distance d. Any point not meeting these criteria is a candidate outlier to the rest of the dataset. Detection performance depends on user-defined parameters, distance metric selection, and the dataset. This method makes no prior distribution assumption but assumes time invariance. Additionally, DBOD performance suffers in high-dimensional data due to its reliance on the distance metric, the limit of which tends to zero in medium-to-higher dimensions ($k \geq 5$). This is also known as the curse of dimensionality. Its complexity $O(\cdot)$ is a quadratic of the dataset size n [80] and the number of dimensions k and can be expressed as $O(n * log(n) * k)$ [81] or $O(n^2 * k)$ in the worst case. Several innovative approaches were developed to adapt to these challenges by tackling computational cost and execution time [82,83], using simple random subsampling (SRS) prior to OD, reducing the dimensionality and estimating a probability density function over the data [84]. The triangle OD (TOD) [83] uses geometric reasoning based on a dissimilarity matrix on the distance to a collection of neighbors to select the decision thresholds. Context-aware distance (CaD) [85] is used for trajectory description in video analysis.
- Density-based OD (DSBOD) is an unsupervised ML method that considers the local neighborhood density of observations and calculates the degree of a tuple p in the

d-dimensional dataset of being an outlier relative to its neighbors, also known as the LOF [86,87]. The LOF was first introduced during the International Conference on Management of Data and it detects local outliers based on LDE and the LDF mentioned above. The LOF is a continuous value representing the degree to which a point p is an outlier to the object with c data points, so that the larger the LOF, the greater the risk of being an outlier. The LOF, rather than being a binary classifier of outlier or not, is a continuous value that provides a degree of "outlierness" of each observation, allowing it to adapt to various scenarios, especially for non-linear systems [88]. It is however computationally intensive given that the LOF should be chosen iteratively for all points. Furthermore, because this method assumes access to the full data space and requires multiple passes over the data, it is difficult to use in streaming data. Several LOF adaptations are proposed to address its shortcomings. The incremental local outlier factors (iLOFs) [89], for instance, are calculated for each new data point and incremental multi-class outlier detection (iMCOD) [90] proposes a multi-class OD, while the cube-based incremental local outlier factor (CB-ILOF) [91] utilizes a 3D slice of the data before performing the detection. The distributed local OD in big data (DLOF) [92] leverages distributed computing and storage for improving memory and time efficiency. The method in [93] constructs a weighted index using information entropy to improve accuracy and memory management in real-time high-dimensional data. DSBOD methods are more accurate in the detection of outliers in high-dimensional and contiguously distributed data compared to DBOD, the performance of which degrades as dimensionality increases. The computational complexity function O in a k dimensional dataset of size n is represented as $O(n * log(n) * k)$ for $k \leq 5$, or $O(n^2 * k)$ otherwise [94].

- Clustering-based OD (CBOD) finds its roots in the LOF defining the possibility of local outliers as opposed to global ones. This unsupervised ML method was developed under the assumption that in a dataset, observations that relate to each other are spatially clustered [95]. Its execution follows a two-step process consisting of identifying clusters in data and flagging observations or groups of observations according to whether they are outliers or not. The method also assumes that an observation might be an outlier relative to a local cluster even though, globally, it might seem to be normal. One of the most popular clustering algorithms is density-based spatial clustering of applications in noise (DBSCAN) by Ester et al. [96], which uses the dense property of clusters. Its predecessor, CLARANS (Clustering Large Applications based on RANdomized Search), uses K-medoid clustering that is memory-intensive and impractical for large datasets as all data and objects are manipulated in memory during its entire execution. DBSCAN instead employs the concepts of density reachability and density connectivity, which define the connectedness of points through their neighbors, while a single user parameter k defines the minimum number of points required in each cluster. CBOD is advantageous for OD in dense datasets compared to DBOD. It is similar to DSBOD but has the advantage of identifying global, collective, and contextual outliers (a point outlier in the context of a cluster). In higher dimensions, the data become sparse and affect clustering performance. Its computational complexity is a quadratic function of the dataset size n, and of the number of dimensions or features k involved $O(n^2 * k)$ [97].
- Angle-based OD (ABOD) was first introduced during the International Conference on Knowledge Discovery and Data Mining [98] and addresses the performance disadvantage of OD in high-dimensional data. Instead of assuming an association of observations based on spatial distance, this type of method assesses the variance of the angle between vectors of a point to others to calculate their proximity. ABOD is a non-parametric method, which is more efficient at OD in higher dimensions than DBOD and DSBOD, which rely on distance metrics. However, it is slower than its alternatives on larger datasets given that its complexity is cubically related to n [98,99]. It does not account for temporal/sequential correlation in streaming data and is time-

consuming to execute. Applying a sample size reduction method might help improve its performance. There are improvements to ABOD in the literature, such as FastABOD, which is faster in low dimensions with large datasets, data stream angle-based OD (DSABOD) [100], and angle-based intrinsic dimensionality (ABID) [101], which improves the speed of detection by applying dimensionality reduction.

- Support Vector Machines (SVMs) were introduced by [102] and is a supervised ML technique that aims to find hyperplanes that divide any dataset represented in an n-dimensional hyperspace into spatially separated classes with the maximum margin between them. It relies on linear functions for linearly separable data, and on kernel functions otherwise [103]. The method is originally supervised, designed for pairwise comparison, and is nonprobabilistic. It is suitable for classification tasks, and therefore, popular for OD problems; however, this makes it impractical for unlabeled streaming data. One-class SVM (OC-SVM) is an adaptation of SVM allowing unsupervised OD by training the model to learn normality in data [104] and detect deviation or novelties in new data. This method is used for OD in WSN [105], 5G IoT [106], or in combination with deep learning methods for unsupervised feature extraction and learning [107]. Online SVMs have also been developed to incrementally update the SVM models on unseen data [108] but may be computationally expensive, especially when using kernel functions. SVM does not account for temporality or sequencing [109] in data, and therefore requires a combination with other time-agnostic models for anomaly detection in temporal or sequential data. This method operates optimally on high-dimensional datasets with a small number of samples.

In summary, most articles focus on either the computational complexity of the OD methods or on their performance on benchmark datasets in terms of accuracy or effectiveness at detecting outliers. Table 3 provides an overview of how well-suited each method covered in this section is for the detection of specific types of outliers, with optimal dataset size n and dimensionality k, to yield the best OD performance. This table contains the following seven columns:

- Method: indicating the OD being described.
- Concept: containing the general concept the method is based on.
- Parametric: indicative of whether the method is parametric or not.
- Type of outliers: outlier type suitability detected by the method.
- Size: the optimal data size to implement the method.
- Dimensionality: the optimal dimensionality supported by the method.
- Computational Complexity: A function indicating the training complexity.

Table 3. Characteristics and computational complexity of ML-Based OD methods.

Method	Concept	Parametric	Type of Outliers	Data Size	Dimensionality	Computational Complexity
DBOD	Distance	Yes	Global	$n \geq 5000$ or 10,000	Low	$O(nlog(n) * k)$ or $O(n^2 * k)$ [84]
DSBOD	Density	Yes	Global Contextual	$n \geq k+1$	Low	$O(nlog(n) * k)$ or $O(n^2 * k)$ [97]
CBOD	Cluster	Yes	Global Contextual Collective	$n \geq 10{,}000$	Low	$O(n^2 * k)$ [100]
ABOD	Angle	No	Global	Small to medium	High	$O(n^3 * k)$ [101,102]
SVM		No	Global	Small	High	$O(n * k)$ Linear $O(n^2 * k)$ or $O(n^3)$ Kernel function

In brief, considering the ML algorithms, the ABOD and SVM are the only methods that offer the ability of OD in high-dimensional data, but both come with a greater computational complexity as the dataset size increases (cubical) and in the case of non-linearly separable data for SVM. DBOD, DSBOD, and CBOD exhibit the opposite behavior as they are more suitable for OD in low-dimensional data, are less complex, and perform optimally at a reasonably large data size. It is generally recommended to have an n that is five to 10 times larger than k for DBOD, and an n greater than the number of points p per cluster C for CBOD. Additionally, ABOD is the only parameter-free method in this category, and it offers more flexibility for OD on data with no empirical parameters often needed to initialize other methods. Table 3 also shows that OD methods relying on local factors are the best option when it comes to contextual OD, albeit at a higher computational cost in lower-dimensional data. Most of the ML methods can be tuned to support online learning but at a much higher computational cost. Online SVM and incremental SVM are variants of SVM used for OD tasks in the literature.

3.3. DL-Based Methods

DL was originally introduced in 1957 [110] and was initially called a perceptron. It has recently regained substantial interest from the scientific community, fueled by the lower price of computing infrastructure such as processors, memory, and storage, increased internet availability and throughput, and the fast-growing cloud services market, which democratizes access to seemingly unlimited computing resources from anywhere in the world. DL is based on the Artificial Neural network (ANN), which is the basic NN model, representing a multi-layer meshed network of neurons, which itself is a computational unit representation of a perceptron [111]. One major difference between DL and other methods previously discussed for OD that it is a model-based approach, whereby a model needs to be trained and then used for OD. Training NN models is a non-trivial exercise as they are data-greedy, computationally intensive, and time-consuming, which leads to inefficiency when it comes to OD in streaming data. However, after the training, the model can be used offline on unobserved data and in a parallel or distributed way to speed up the detection. It could also be retrained if the dynamic of the data evolves, making it a good candidate for handling concept drift and evolution. There are two major types of DL models, **generative models** that learn from input data to generate a joint probability distribution dataset with fewer dimensions and **discriminative models** using the conditional distribution of all hidden variables in the data [112]. The main types of NN models are described below:

- Multi-layer Feed Forward (MLF) is one of the most prominent NNs. It comprises sequentially cascaded layers, each neuron of which receives the output of previous layers as input [113]. These networks may be fully connected, where predecessors' outputs are fed as input to all successors, or partially connected. This method uses back-propagation as an error correction algorithm based on the partial derivative of the output error function $E[\cdot]$ to update the weights and the thresholds of each of the previous layers in the cascade. MLF is used for OD in the field of network intrusion detection [114], anomaly detection in IoT networks [115], and in multi-sensor systems [116]. A drawback of MLF in performing OD on streaming data is the fact that it requires labeled data for the model's initial training before OD. This implies that there is empirical labeled data for model pre-training, and this is not always the case in real-world applications, unless online training is used, which would come at a computational cost of $O(n * k * l)$, where l is the depth of the NN. This notation will be used throughout this section for other DL methods. A compromise is to use the pre-trained model, relying only on online transfer learning (OTL) [117] to retrain the model in case of changes in the data stream structure (concept evolution), but this comes at a computational cost of $O(b * k)$, where b represents the subset of new points considered for training. This method is supervised as it requires labels in the training data.

- Recurrent Neural Networks (RNNs) were introduced by Elman, J.L. in 1990 [118] to address the limitations of MLF NN in dealing with temporally dependent data, such as time series and sequential data [119], by introducing a feedback mechanism that allows a neuron output to be fed as input back to itself in hidden layers [120] or to preceding neurons [121]. This allows them to process sequences of data with variable length, therefore representing an excellent solution for dealing with streaming data. This method is typically used on temporal data such as time series and streaming data but is limited by the number of lags it can feedback to, making it a short-term memory network. Long short-term memory (LSTM) and gated recurrent unit (GRU) are the two most used variations of RNNs [122], which are extensively used in the literature for solving many problems involving real-world data. RNNs are better adapted for contextual outliers given that detection is achieved within the context of short-term lags at a cost of a cubical computational complexity, similar to the MLF NN. OTL is also very frequently used with RNNs for OD in streaming data. RNNs are used in environmental science [123], information security (biometric authentication) [124], and sensor networks [125].
- Long Short-term Memory (LSTM), developed by Hochreiter, S. and Schmidhuber, J. in 1997 [126], is a tweak to the RNN that allows keeping the long-term memory of past lags in the data, thus addressing the RNN's weakness in processing long-term sequential and temporal data. LSTM introduces a three-gate system with an input gate, an output gate, and a forget gate, allowing newly acquired information into the memory cell to be memorized or forgotten. One important pattern in the literature is that LSTM appears in most research related to anomaly detection in time series using NNs. Even when not employed as a direct method for detecting outliers in the spatial domain, it is combined with other OD models that have higher efficiency in the same, while LSTM covers the temporal aspect of detection by taking advantage of its long-term memory ability. One of the advantages of LSTM is that it maintains information longer using branching, which helps to reduce the vanishing gradient problem. A notable drawback of this method is that, similarly to the RNN, it processes the information sequentially [127]; therefore, it cannot utilize the computationally efficient parallel processing offered by the graphical processing units (GPUs). Some authors combine LSTM with RNN for anomaly detection [128,129] or leverage online learning transfer [130–132] for faster retraining to improve detection performance. Model pretraining requires labeled data.
- Autoencoders (AE) are special types of symmetrical [133] and unsupervised [134] NNs that use the input data as the target output, using its encoder layers to reduce the dimension of the input data while the decoder layers reconstruct it. It is an alternative to the dimensionality reduction algorithms that use both linear and non-linear transformations to reduce the data dimension as opposed to principal component analysis (PCA), which relies solely on linear models for feature extraction. This comes very handy when dealing with real-world data streams with a large number of features as it would help with escaping the curse of dimensionality faced by other ML algorithms in high-dimensional data. Additionally, this method does not require labeled data because the input data are used as output. This makes AEs an interesting candidate for dealing with the unpredictable nature of real-world data streams with often frequently changing structures and content over time.
- Convolutional Neural Networks (CNNs) are prominently applied in image classification tasks [135] and are designed to accept tensor data at the input layer and use their hidden layers to extract features from the tensor. At the end, it returns a result that corresponds to the specific goal at the output layer. CNNs are, in general, composed of three common building blocks [136], the first of which is the convolutional layer that employs the convolution of filters to compute a feature map of the input by using either the sliding sum of 2D filters/3D filters or by matrix multiplication [137]. Then, it is the pooling layer [138], which applies summarization functions, such as maximum

or average, on the output of previous layers to produce a lower-dimensional matrix as the output. Finally, there is the fully connected layer that performs the classification, defined as a function f of the sum of product of a $j \times i$ dimensional weight matrix $W^{j \times i}$ (of j rows by i columns), by x^i input features and the m dimensional bias matrix b^j, represented as $f\left(W^{j \times i} x^i + b^j\right)$. An advantage of the CNN is that it can be trained on a small sample of high-dimensional data. However, the training complexity is very high and may affect its performance in the streaming data context. Transfer learning is generally used in this case as the capabilities of a model trained on a very large dataset are transferred to a smaller dataset for OD.

- The Deep Belief Network (DBN) is a probabilistic generative model [139], formed by stacking Restricted Bolzmann Machines (RBMs) and designed to face the challenge of NNs overfitting at the learning stage due to poor parameter selection, which leads to increased data greediness. RBM was introduced in 2006 by Geoffrey Hinton and is composed of two layers: a layer of visible i units and a layer of j binary hidden units h, where the total energy $E(\cdot)$ of the machine is calculated as follows [140]: $E(vh) = -\sum_{I,j} v_i h_j W_{ij} - \sum_i v_i a_j - \sum_j h_j b_j$. The DBN uses a layer-by-layer training approach also known as RBM unsupervised training, as well as error back-propagation for fine-tuning [141]. DBN's advantage for OD in streaming data is its ability to handle high-dimensional data and perform feature extraction [142], which significantly improves prediction performance. It also allows assigning a probability of outlierness to each outlier, which is very useful for setting the appropriate decision threshold.
- Generative Adversarial Networks (GANs) represent a dual network composed of a generative unit and a discriminative unit, which was initially introduced by Ian Goodfellow et al. [143] in 2014. These architectures are based on the concept of learning normality from an input dataset, using the normal probability distribution of the data input in its generative network to produce similar data, and then using its discriminative network to identify original data from the output [144,145].
- Transformer NNs, introduced by Vaswani et al. in 2017 [146] in the article "Attention is all you need", are among the most prominent topics in the DL field at present, as they provide more possibilities in various fields, especially in natural language processing and computer vision. Transformers rely on a state-of-the-art method called the attention mechanism. One of the key advantages of transformers over other NN models such as RNN and LSTM is their ability to parallelize processing by taking multiple inputs versus the prominent sequential approach, hence improving model training and computational performance. Additionally, the multi-headed attention layer allows the NN to focus only on important features of the hidden layer output by applying scoring, according to which features of high importance are emphasized while the influence of others is diminished. This addresses the vanishing gradient problem that impacts the performance of RNN and LSTM in the case of, for instance, neural machine translator systems.

4. Applications of OD Methods

This section summarizes applications of the methods grouped by the three main broad categories, Statistical, ML-based, and DL-based, in Table 4. This table is straightforward and has only three columns. The broad categories listed above contain abbreviations of the methods as defined in the relevant sections, and the applications by industries. In addition, the Table leads to conclusions according to the important results and challenges presented in the reviewed literature.

Table 4. Classification of the OD methods by their applications and broad category.

Broad Category	Method	Applications
Statistical-based	GBOD	Transportation [53], medicine [56,58], energy monitoring [147], resource management and orchestration in wireless networks [148], and Cloud Systems [54,57] for log analysis.
	BPOD	Energy monitoring and power consumption [149]
	RBOD	Information technology, fault detection in the energy industry [150], and environmental science [151].
	MAD	Environmental science for atmospheric data analysis [152], water level monitoring [153], and sensor calibration [154].
	CBOD	Child nutrition monitoring in healthcare [66], wind turbine power monitoring [68], and process monitoring in computer science [70].
	GMM	Sensors in civil engineering [73], streams clustering for customization of IoT [155], and more general networks [156].
	HBOD	WSN data analysis [157].
	KBOD	Denial of service detection and analysis in computer networks [158], intrusion detection [159], and nuclear security [160].
ML-based	DBOD	Knowledge discovery and pattern recognition [82,83] and anomaly detection in video streams [85].
	DSBOD	Intrusion detection in computer networks [87] and physical layer security in spectrum sensing for customization in CR networks [161,162].
	CBOD	Computer networking [163] and orchestration of WSNs [164,165].
ML-based	ABOD	Oil and gas [166] and cyber security [167].
	SVM	Threat detection and prevention in 5G IoT [106] and WSNs [109] and Vehicle Ad Hoc Networks (VANET) [104].
DL-based	MLF	Network intrusion detection [114], intelligent transport systems [168], medicine, and customization of IoT and WSNs [115,116,169].
	RNN	Environmental science (water quality monitoring) [123], biometric authentication [124], video surveillance, sensor data reconstruction [125], malicious insider threat, network traffic, and electricity theft detection.
	LSTM	Self-organization and customization of cellular [128] and computer [129] networks, renewable energy, and intelligent transportation systems [132].
	AE	Customization of IoT networks [170], smart farming in agriculture [171], aerospace industry, medical field, environmental science, and WSNs in maritime [172].
	CNN	Cyber security [173], in-vehicle networks [174], and anomaly detection in medical imagery [175,176].
	DBN	Data analysis from WSNs [177], industrial systems, and intrusion detection systems in cyber security [178].
	GAN	Anomaly detection in medical imaging [179], data mining and knowledge discovery, customization of IoT networks [9], telemetric data, and radio spectrum reconstruction [180].
	Transformer NNs	Aerial video streaming [181], multivariate OD in IoT [182], power grids and water distribution industries, processing time series data, events sequence content-aware anomaly detection [183], medical imagery on electrocardiogram (ECG) images, and vibrating signals and remote sensing [2,184].

The statistical methods have been used in many problems relevant to telecommunication networks and systems [155–158], analysis of physical phenomena, or human medical conditions [56,58,62,66,68,70]. This is due to the existence of sufficiently developed assumptions on data distributions (in the case of parametric models), which have been tested and verified on empirical data collected for the various related applications. This

allows the development of OD models using historical data for the known distributions and implies lower computational complexity and faster OD. This category also includes examples of applications [66,157–160], for which non-parametric methods were used to increase detection accuracy when a priori knowledge is unavailable. Thus, they identify and predict the anomalies caused by outages or other undesirable events, even at the cost of reduced energy efficiency. The statistical methods are generally difficult to compare due to the variety of their performance parameters, which are largely dependent on the data as they are derived from statistical properties (such as mean, kurtosis, etc.). Thus, the choice of a particular method is to be data-driven and based on preliminary analysis of the available samples, while keeping in mind the application scenario. For example, if the algorithm is to process unobserved data in real-time, then less computationally intensive methods (such as KBOD, HBOD, COPOD, and CBOD) may be used, whereas when the goal is a more precise offline analysis of data for OD model development and the application of new data, MAD and GBOD methods should be applied. For large multivariate datasets, CBOD and GMM are options to consider for dimensionality reduction.

The ML-based methods for OD often have increased computational complexity due to their non-reliance on input data distribution assumptions, hence the need to compute specific features (such as distance, density, and the angle between the observations) from the data before starting the actual detection. This creates additional challenges for their application in streaming data and their complexity increases with data dimensionality. If the input is of low dimensions, then DBOD methods may be applied with adequate efficiency, for instance in slow-motion video streams [85], or DBSOD methods in the case of resource management and orchestration in a CR network with a small number of nodes [87,161,162]. When substantial computing resources are available for offline training, CBOD or ABOD are good candidates for distinguishing between multiple classes of users (also between incumbent and malicious) in large-scale datasets with many connections among themselves (such as computer or wireless networks) [66,68,70,166,167]. In these cases, the method's efficiency is often determined by the number of samples (usually in the order of tens of thousands or more) available for training. The ML-based methods may be more adequately compared to one another as they are often applied in classification problems, thus sharing common KPIs such as probability of detection, true positive rate, etc.

Due to the substantial growth of DL in recent years, the methods in this category are naturally the most numerous and diverse. They include applications in wireless network customization and self-organization (cellular/IoT/WSN/vehicular communications), medical imaging and diagnostics, remote sensing of physical phenomena, video surveillance, and others. Depending on whether the NN needs to predict/reconstruct the input sequence or classify its type, the choice is usually made between transformers, AEs, GANs, CNNs, MLF, RNNs, and LSTMs. These methods, similarly to ML, require pretraining but on much larger datasets (hundreds of thousands of samples) than for ML (thousands). Therefore, they need very substantial computing resources (such as multiple GPUs). However, if the model has (1) adequate depth to its architecture (which is also data-dependent), (2) appropriately chosen parameters (usually determined empirically or estimated using additional data), or (3) a large enough dataset (for features learning), then it may achieve high accuracy on new observations very quickly using little computational power. As observed from Table 4, the CNN, LSTM, RNN, and DBN models are most often used for telecommunication systems, although not solely for such. Very few studies using Transformer NNs in general and particularly in the field of telecommunications are available, due to it being the current state-of-the-art in the field of NNs. The ability of DL models to accept a large variety of input data types makes them an easier and popular choice for multiple applications. Comparing their performance is rather straightforward as they can model the output KPIs, such that they are relevant to the input, i.e., either reconstruction or classification accuracy.

5. Review of Datasets for OD in Streaming Data

Identifying datasets for training and testing OD models is a challenging endeavor. Table 5 summarizes prominent datasets found in the literature and categorizes them into the OD methods reviewed in this paper. The Table includes the following columns—Dataset Name (the datasets' online titles), the Year of publication or creation, the Number of Features in the dataset, its Size, Application, the Method it is used for, the References that use it, and its Source (institution or organization). Fields in which there are no applicable data are denoted as N/A.

Researchers often use empirical data to build models, which are then applied to new data for outlier detection. Some studies use live data collected by users for real-time analysis or due to the lack of existing data [160]. For instance, in telecommunications studies, data might be collected in real-time by users [162,164] or through existing sensor networks [152,179]. Public datasets such as IoTID20, Corel Histogram, and BOT-IOT for IoT-related studies and KDD99, ISCX NSL-KD, and CICIDS2017 for network intrusion detection are also commonly used.

These datasets are often available in databases such as the OD datasets (ODDS) database [185], IEEE Dataport [186], and the websites of university research laboratories [187]. These sources offer extensive historical data, which need to be organized and sequenced to simulate streaming data during analysis or streamed using relevant tools. For real-time streaming data, researchers may need to collect their own data using sensors or tap into online data sources such as social media via application programmable interfaces (APIs), which are often rate-limited or not free. In the field of telecommunications, real-time data is hard to find, leading researchers to rely on historical data or collect data in rea-time using sensors.

Table 5. Review of prominent datasets for OD.

Dataset Name	Year	Number of Features	Size	Application	Method	Reference	Source
KDD99	1999	41	58.3 k	Intrusion detection dataset	DSBOD GAN	[87,162, 179]	University of California Irvine [188]
ISCX NSL-KD	2009	41	125 k	Intrusion detection	GAN	[179]	University of New Brunswick [189]
IoTID20	2020	85	625.7 k	IoT Intrusion Dataset	Ensemble	[156]	IEEE Data Port [190]
Corel Histogram	2024	31	68 k	IoT in smart cities	DBOD	[82]	ML Pack [191]
DS2OS Traffic Traces	2018	14	322 k	IoT in smart cities	MLF	[169]	Aubet, FX [192] available on Kaggle
Wireless Sensor Network	2004	8	2.5 M	WSN	CBOD	[69]	Intel Berkeley Research Lab [193]
A Poisson–Gaussian Denoising Dataset	2018	N/A	N/A	Image denoising	CBOD	[71]	Zhan Y. et al. [194]
HAR	2013	562	10.3 k	Human activity recognition	DBN	[164]	University of California Irvine [195]
DSA	2012	315	9.1 k	Daily sport activities	DBN	[164]	University of California Irvine [196]
GAS, GT, IR, WM	2004	N/A	N/A	Comparative study	RBOD	[44]	Kaggle

Table 5. *Cont.*

Dataset Name	Year	Number of Features	Size	Application	Method	Reference	Source
SCADA data	2011	8	N/A	Wind turbine fault detection	RBOD	[137]	Collected by user
Ultrasonic sensor	2019	N/A	N/A	Monitoring water level and discharge	MAD	[140]	Collected by user
Tropospheric Data Acquisition Network	N/A	9	N/A	Sensor data on atmospheric temperature	MAD	[139]	Center for Atmospheric Research [197]
Synthetic and testbed	2018	N/A	N/A	Synthetic and testbed	GMM	[142]	Collected by user
Network traffic	2017	N/A	N/A	Real-time data	GMM	[143]	Collected by user
CICIDS2017	2017	83	25 users in 5 days	Intrusion detection evaluation	KBOD	[145]	Canadian Institute for Cybersecurity [198]
Portable radiation spectrometer	2021	N/A	N/A	Real-time data	KBOD	[147]	Collected by user
Cognitive Radio Network	2021	N/A	N/A	Real-time data	CBOD	[149]	Collected by user
Wireless Sensor Network	2023	200	400 s	Real-time data	CBOD	[151]	Collected by user
Network Forensic Analysis	2019	35	72 M	BOT-IOT	GAN	[166]	Cyber Range Lab—UNSW [199]
UC-Merced 256 × 256 images	2022	21	100	Remote sensing scene classification	CNN and Transformer	[172]	University of California, Merced [200]
AID (600 × 600)	2019	17	3 k	Remote sensing scene classification	CNN and Transformer	[172]	AID scene from Wuhan University [201]
NWPU-RESISC45 (256 × 256)	2021	12	31.5 k				Northwestern Polytechnical University dataset [202]
OPTIMAL-31 (256 × 256)	2018	31	186 k				Hyperspectral Image Dataset [203]

6. Summary of Assumptions and Outlier Types

In this section, Table 6 clarifies the basic assumptions of various outlier detection methods to better understand possible challenges related to their application to streaming data. The Method column lists the major OD methods, the Distribution Assumptions column indicates whether the model assumes that the data have a particular distribution, the Data Types they are more suitable for, and the Outlier Types they are more efficient at detecting. The description of the actual detection approach is in the Outlierness column, and finally the Method-specific Assumptions are listed in the column with the same name. The following abbreviations are used in the Data Type and Outlier Type columns—N is

for Numerical, N(C)—Numerical Continuous, N(D) is for Numerical Discrete, C is for Categorical, T is for Temporal (Sequential), Ts denotes a tensor, Tx—Textual, G is for Global, C is for Contextual, CL is for Collective, and "All" denotes G, C, and CL. Additionally, the online learning column presents the ability of a method to learn offline (Off) or online (On). Whenever the asterisk (*) is used with the label Off, it indicates that the method can be adapted to online learning. The supervision column indicates whether a method requires labeled data during its training phase, which makes it unsupervised (U) or supervised (S).

Table 6. OD methods categorized by assumptions and related outlier types.

Broad Category	Method	Distribution Assumptions	Data Type	Online Learning	Supervision	Outlier Type	Outlierness	Method-Specific Assumptions
Statistical-based	GBOD	Normal	N(C)	Off *	U	G	Distribution-based	Univariate and mean centered
	BPOD	N/A	N(C)	Off *	U	G	Distribution-based	Univariate symmetrical
	RBOD	Normal	N | T	Off	S			Normal, Stationary
	MAD	None	N	Off	U	G	Distribution-based	Univariate Symmetrical
	Copula-based OD	None	N(C)	Off *	U	C	Distribution-based (learned)	A joint distribution can be built from marginals.
	GMM	Normal	N(C)	Off *	U	C | G	Distribution-based	Same family of univariates
	HBOD	None	N(C | D)	Off	U	G	Distribution-based (learned)	Data distribution can be learned
	KBOD	None	N(C)	Off *	U	C	Density	-
ML-based	DBOD	None	N(C)	Off *	U	G	Spatial closeness	The spatial distance can be calculated.
	DSBOD	None	N(C)	Off *	U	G | C	Spatial closeness and density	Observations are spatially contiguous and densely distributed.
	CBOD	None	N(C) | C	Off *	U	C | CL	Spatial closeness	Outlierness relates to local clusters.
	ABOD	None	N(C)	Off	U	G	Angular closeness (Trigonometric)	Does not assume spatial projection.
	SVM	None	N | C | Tx | Ts	Off *	S	G	Class boundary or Distribution-based	Data can be separated by class
DL-based	MLF	None	N/C/Tx	Off	S	G | C	N/A	-
	RNN	None	T	On	S	C	N/A	Short-term autocorrelation.
	LSTM	None	T	On	S	C	N/A	Long-term autocorrelation.
	AE	None	N/C/Tx	On	U	G | C	Encoding/decoding ability	No linear assumption
	CNN	None	Ts	On	S	C	N/A	-
	DBN	None	N/C/Tx	On	S	G | C	N/A	Data are hierarchical.
	GAN	None	N/C/Tx	On	U	G | C	Distribution-based (learned)	Data distribution can be learned
	Transformer	None	Ts/C/T	On	S	All	N/A	-

After reviewing applications of OD methods in various industries, the basic assumptions of each of the methods are summarized to provide guidelines for the preference of one method over the other for the OD task in streaming data. Table 6 below summarizes the assumptions by method together with the data types they are restricted to and the type of outlier they are generally used to detect. Additionally, the overall mechanism of outlierness determination used by the respective method is mentioned in this Table. The spatial coverages of the detection method, as well as some non-distribution assumptions specific to the methods, are also listed. In Table 6, OD methods are grouped into broad categories (Statistical, ML, and DL), and then further classified depending on whether they are based on their underlying distribution or a specific assumption.

Some of the statistical methods (BPOD, MAD-OD, COPOD, HBOD, and KBOD) and all ML and DL methods can be considered distribution-free because they generally make no prior assumption on the input data's underlying distribution and use learning on empirical data to estimate the distribution parameters, before modeling new data and detecting outliers. This makes them very potent for OD in real-world data with unknown distributions, subject to concept drift or evolution. Statistical methods support mostly numeric continuous data except for HBOD, which is suitable for discrete data while RBOD supports temporal data (time series). All ML methods use numerical continuous data except for the CBOD methods, which are also applied to categorical data. DL methods are applied to a variety of data types including numerical, categorical, and textual. However, RNN and LSTM are the most often used in the literature when it comes to temporal data. Transformers, which represent the most recent of the methods, are applied to OD in sequential data, especially in the areas of text analysis and Large Language Modeling (LLM).

In relation to the types of outliers optimally detectable by the various methods, for collective outliers, CBOD and Transformers are the most prominent ones. Contextual outliers are better detected by methods relying on a joint distribution (RBOD, COPOD, and GMM-OD), on the density (KBOD, DSBOD, and CBOD), and by all DL methods. When it comes to high-dimensional data, COPOD and GMM-OD from the statistical methods, and AE from the DL methods, are generally preferred, while CNN is generally used for feature extraction. The GAN, AE, copulas, and GMM methods are used for data augmentation in the case of a small sample size.

7. Classification of OD Methods by Predefined Criteria

7.1. Criteria Definition

Several papers compare OD methods based on their performance on benchmark datasets or on data from specific domains of application and use model performance metrics to classify them in terms of their detection capabilities. This Section outlines the criteria for assessing OD in the context of streaming data, considering its major characteristics, processing mechanisms, and challenges mentioned earlier in this paper. Table 7 summarizes the following six relevant criteria:

- The sample size n requirement for the method performance in OD in relation to the selected window mechanism is referred to as data greediness; a model considered data greedy will be marked with "Yes", otherwise with "No".
- The optimal data dimensionality to perform OD—this will take values of Low if the model performs better in low dimensions $k \leq 5$, in medium dimensions of $5 < k < 10$, or high dimensions of $k \geq 10$.
- The computational complexity of the method, which is defined as a function of the sample size n, dimensionality k, and depth l of the model used in the method; given that $O(\cdot)$ is the complexity function, a model will be classified as:
    - Low complexity if O is a linear function of n and the data dimensionality d.
    - Medium complexity if O is quadratic for n and linear for k.
    - High complexity if O is a quadratic function of both n and k and uses the model depth l or the number of hidden layers h as parameters.

- The ability to detect outliers in temporal data is called temporal ability; a model will be marked as "Yes" for its temporal ability, and "No" otherwise.
- The flexibility and adaptability of the method are defined based on the number of parameters used by the method or whether it is supervised, semi-supervised, or unsupervised and will be classified, respectively, as low, medium, highly flexible, or adaptable; a model considered flexible will be denoted with "Yes", otherwise "No".
- The model's robustness is indicative of whether the model is sensitive to outliers or not. A model considered robust is denoted in Table 7 as "Yes", and as "No" if it is not.

Table 7. Definition of criteria for OD methods.

Criteria	Definition
Data Greediness	Referring to the amount of data required for the method to perform OD task.
Data Dimensionality	Indicative of the dimensionality on which models would optimally perform OD.
Computational Complexity	This relates to training complexity O as a function of the dimensionality d, n, and l: • Low: O is only function of n~$O(n)$. • Medium: O is function of at least 2 parameters—$O(n, d,$ or $l)$. • High: O if function of more than 3 parameters—$O(n, d,$ and $l)$.
Temporal ability	Defined as: • Yes: Can handle OD in temporal data. • No Cannot directly handle OD in temporal data or requires combination with another method to handle temporality in data (ensemble methods).
Flexibility and Adaptability	Defined as: • Low: Method is supervised (requires user input), requires more than 1 parameter, can handle multiple data types, can adapt to data changes (concept drift or evolution), does not support online learning. • Medium: Method is semi-supervised, requires only 1 parameter, relies on model that requires retraining upon data change, does not support online learning. • High: Method is unsupervised, parameter-free, distribution-free, or adapted for OD on multiple data types, supports online learning.
Robustness	Accounts for whether the model is sensitive to outliers: • Yes: The model is sensitive to outliers. • No: The model is not sensitive to outliers.

7.2. Classification of OD Methods by Predefined Criteria

Selecting the appropriate OD method is non-trivial, especially in a streaming data context, due to the complexity and possible variability that such data naturally carry. To facilitate the selection of an appropriate method, Table 8 classifies the OD methods reviewed in this paper based on the set of criteria defined in Section 7.1. This Table is intended to provide guidelines for OD method selection based on a comprehensive set of criteria. Because the studies surveyed in this article used different datasets, performance metrics such as accuracy, precision, recall, and $F1$ score are not included in the Table. Instead, it groups OD methods into broader categories (statistical, ML, and DL), which are broken down into parametric and non-parametric sub-groups for statistical-based methods, proximity- and deviation-based methods for the ML category, and generative, discriminative, or both for the DL methods. Then, all of them are classified based on their data greediness, where most statistical methods are considered non-greedy except for KBOD, COPOD, and GMM-OD, which require enough to understand the marginal properties and build a joint probability distribution based on them. DL methods are the most data-greedy, except for CNNs which may rely on much fewer inputs for learning, or generative models which are more frequently used to augment data size and diversity in the case of small datasets. Among the ML methods, DSBOD and CBOD are the most computationally intensive. When a method is classified as a high-dimensionality method, this denotes that it is capable of

handling raw high-dimensional data. The data themselves are classified as low-, medium-, or high-dimensional based on the dataset's number of features k. Among the methods, COPOD, GMM-OD, ABOD, CNN, AE, and Transformers are those marked as most suitable for high-dimensional data. Then, the computational complexity column provides a scale of complexity as a function of the sample size, dimensions, and model depth as applicable, while the complexity class column provides a ranking of the complexity, which is based on the data within the computational complexity column. In the temporal ability column, the ability of the model to process temporal changes in the data is denoted as "Yes" for methods that have such an ability, and "No" for those that do not. The concepts of flexibility and adaptability explained in Section 7.1 are covered in the column with the same name. Finally, the robustness of the model indicates its sensibility to outliers.

Table 8. Classification of OD methods by predefined criteria.

Broad Category	Type of Methods	Method	Data Greediness	Data Dimensionality	Computational Complexity	Complexity Class	Temporal Ability	Flexibility and Adaptability	Robustness
Statistical-based	Parametric	GBOD	No	Low	$O(n * log(n))$	Low	No	Yes	No
		BPOD	No	Low	$O(n * log(n))$	Low	No	Yes	Yes
		RBOD	No	Low	$O(n * di + dj)$	Medium	No	No	No
		MAD	No	Low	$O(n * log(n))$	Low	No	Yes	Yes
		COPOD	Yes	High	$O(d * nlog(n) + d2 * n + d3)$	Medium	Yes	Yes	Yes
		GMM	Yes	High	$O(t * n * d2 * k)$	High	No	Yes	No
Statistical-based	Non-parametric	HBOD	No	Low	$O(n)$	Low	No	Yes	Yes
		KBOD	Yes	Low/Medium	$O(n * log(n) * 2)$	Low	No	Yes	Yes
ML-based	Proximity	DBOD	No	Low	$O(n2 * d)$ or $O(n * d * log(n))$	Medium	No	Yes	No
		DSBOD	Yes	Low	$O(n2)$ or $O(n * log(n))$	Low	No	Yes	Yes
		CBOD	Yes	Low	$O(n2)$	Low	No	Yes	Yes
	Deviation	ABOD	No	High	$O(n3 * d)$	Medium	No	Yes	Yes
DL-based	Discriminative	MLF	Yes	Low/Medium	$O(n * (d * h + l * h2))$	High	No	Yes	Yes
		RNN	Yes	Low/Medium	$O(n * d * t)$	High	Yes	Yes	Yes
		LSTM	Yes	Low/Medium	$O(n * T * (d * h + h2))$	High	Yes	Yes	Yes
		CNN	No	High	$O(n * d * k2 * f)$	High	No	Yes	Yes
	Generative	AE	No	High	$O(d * h + h2)$	High	No	Yes	Yes
		DBN	No	Low/Medium	$O(n * k * l * h2)$	High	No	Yes	Yes
		GAN	No	Low/Medium	$O(z \cdot h + lg \cdot h2)$	High	No	Medium	Yes
	Generative and Discriminative	Transformer	Yes	High	$O(n * d2 * l + n2 * d * l)$	High	Yes	Yes	Yes

8. Discussion

OD in streaming data is a relevant problem for applications in multiple fields, particularly in the telecommunications industry. Depending on the type of outliers, the nature of the problem, the structure of the data, the timeliness of the detection, and the resource availability, various OD methods can be used separately or in combination, with consideration of the computational cost. Popular applications of OD in Telecommunications are in the fields of WSNs, remote sensing, CR networks, and fault detection in the IoT or

special events (such as security incidents and intrusion detection) identification, with the overall aim of building cognitive self-healing systems, especially with the growing need of connectivity fueling the expansion of 5G and beyond wireless networks.

Even though it is often difficult to strictly categorize an OD method for specific requirements, Tables 6 and 8 provide useful information when it comes to key considerations in terms of data availability, type of outliers to detect, and computational efficiency for the selection of an appropriate OD method. DL approaches, prominent in the recent literature, generally require substantive amounts of labeled data and processing power to be trained for detection. They are capable of online learning, except for GAN and AE which can operate in a completely unsupervised manner. ML and statistical-based methods can mostly operate in an unsupervised manner; however, in most cases, they are parametric and require user-defined parameters and tuning. SVM and RBOD methods require labeled data for their training. Considering the computational requirements, statistical-based methods are faster and lighter to operate than ML and DL models and might be recommended for OD in telecommunication applications such as trend analysis, as well as data streams created by one or several sensors in IoT or CR networks. When it comes to advanced applications in telecommunication networks such as pattern recognition, intrusion detection and prevention, sensing, and others, DL methods may be recommended to improve detection accuracy while accounting for changes in data such as concept drift and evolution. A trade-off will have to be made between the computational capability available in the core network or at the edge and the detection speed requirement. For faster detection with low computational requirements, statistical-based methods or pretrained ML and DL might be preferred given that the processing will be faster. Nevertheless, accuracy will be degraded should the data change drastically in real-time if online learning is not implemented.

Several promising research areas have been identified based on this survey. In telecommunications networks with distributed nodes, such as the IoT and WSN, a distributed outlier detection approach combining edge detection with node consensus, similar to blockchain networks, could enhance detection speed and accuracy while being computationally efficient. Another area of interest is exploring the relationship between the computational complexity of various outlier detection methods and different streaming data window mechanisms across diverse benchmark datasets. Additionally, investigating the use of transformer networks for interference management in cognitive radio systems, particularly leveraging online learning to determine the noise floor and distinguish interfering signals from noise, presents another valuable research direction.

9. Conclusions

This article conducts a review of the challenges of OD in streaming data with consideration of the various types of outliers, the data processing challenges with regards to their variety, velocity, and dimensionality, whether they are dynamic, and the computational complexity associated with them. The most prominent OD methods from the broader statistical, ML, and DL categories are categorized according to their major assumptions, types, limitations, and relevant challenges in streaming data. Major applications of these methods in Telecommunications and other industries are reviewed, and the prominence of statistical methods is outlined for methods with available empirical data and a strong theoretical background. Finally, a set of criteria for classifying the OD methods is defined and classification tables are provided and developed, together with guidelines for the selection of the most appropriate methods depending on the specific OD requirements and resource constraints.

Author Contributions: Conceptualization: R.N.M.; writing—original draft preparation, R.N.M. and A.I.; methodology, R.N.M. and A.I.; writing—review and editing, V.P.; verification, P.K., V.P. and A.M.; supervision, P.K. and V.P.; project administration, A.M.; funding acquisition, V.P. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC is funded by the European Union-Next Generation EU, through the National Recovery and Resilience Plan of the Republic of Bulgaria, project № BG-RRP-2.004-0005: "Improving the research capacity and quality to achieve international recognition and resilience of TU-Sofia" (IDEAS) and performed with the support of the Intelligent Communication Infrastructures Laboratory at the "Research and Development and Innovation Consortium", Sofia, Bulgaria.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Wang, Y.; Li, J.; Yang, B.; Li, H.G. Stream-data-clustering based adaptive alarm threshold setting approaches for industrial processes with multiple operating conditions. ISA Transactions. *ISA Trans.* **2022**, *129*, 594–608. [CrossRef]
2. 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]
3. Paul, K.; Chatterjee, S.S.; Pai, P.; Varshney, A.; Juikar, S.; Prasad, V.; Bhadra, B.; Dasgupta, S. Viable smart sensors and their application in data driven agriculture. *Comput. Electron. Agric.* **2022**, *198*, 107096. [CrossRef]
4. Yang, Y.; Ding, S.; Liu, Y.; Meng, S.; Chi, X.; Ma, R.; Yan, C. Fast wireless sensor for anomaly detection based on data stream in an edge-computing-enabled smart greenhouse. *Digit. Commun. Netw.* **2021**, *8*, 498–507. [CrossRef]
5. Juszczuk, P.; Kozak, J.; Kania, K. Using similarity measures in prediction of changes in financial market stream data—Experimental approach. *Data Knowl. Eng.* **2020**, *125*, 101782. [CrossRef]
6. Edge, M.E.; Sampaio, P.R.F. The design of FFML: A rule-based policy modelling language for proactive fraud management in financial data streams. *Expert Syst. Appl.* **2012**, *39*, 9966–9985. [CrossRef]
7. Ma, B.; Guo, W.; Zhang, J. A survey of online data-driven proactive 5G network optimisation using machine learning. *IEEE Access* **2020**, *8*, 35606–35637. [CrossRef]
8. Parwez, M.S.; Rawat, D.B.; Garuba, M. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Trans. Ind. Inform.* **2017**, *13*, 2058–2065. [CrossRef]
9. Ullah, I.; Mahmoud, Q.H. A framework for anomaly detection in IoT networks using conditional generative adversarial networks. *IEEE Access* **2021**, *9*, 165907–165931. [CrossRef]
10. Márquez, D.G.; Otero, A.; Félix, P.; García, C.A. A novel and simple strategy for evolving prototype based clustering. *Pattern Recognit.* **2018**, *82*, 16–30. [CrossRef]
11. ZareMoodi, P.; Kamali Siahroudi, S.; Beigy, H. Concept-evolution detection in non-stationary data streams: A fuzzy clustering approach. *Knowl. Inf. Syst.* **2019**, *60*, 1329–1352. [CrossRef]
12. Chan, H.L.; Lam, T.W.; Lee, L.K.; Ting, H.F. Continuous monitoring of distributed data streams over a time-based sliding window. *Algorithmica* **2012**, *62*, 1088–1111. [CrossRef]
13. Pugliese, L.D.P.; Ferone, D.; Festa, P.; Guerriero, F. Shortest path tour problem with time windows. *Eur. J. Oper. Res.* **2020**, *282*, 334–344. [CrossRef]
14. Blevins, D.H.; Moriano, P.; Bridges, R.A.; Verma, M.E.; Iannacone, M.D.; Hollifield, S.C. Time-based can intrusion detection benchmark. *arXiv* **2021**, arXiv:2101.05781.
15. Yue, W.; Moczalla, R.; Luthra, M.; Rabl, T. Deco: Fast and Accurate Decentralized Aggregation of Count-Based Windows in Large-Scale IoT Applications. In Proceedings of the 27th International Conference on Extending Database Technology (EDBT), Paestum, Italy, 25–28 March 2024; pp. 412–425.
16. Zeng, Z.; Cui, L.; Qian, M.; Zhang, Z.; Wei, K. A survey on sliding window sketch for network measurement. *Computer Networks* **2023**, *226*, 109696. [CrossRef]
17. Baldini, G.; Amerini, I. Online Distributed Denial of Service (DDoS) intrusion detection based on adaptive sliding window and morphological fractal dimension. *Comput. Netw.* **2022**, *210*, 108923. [CrossRef]
18. Iqbal, W.; Berral, J.L.; Carrera, D. Adaptive sliding windows for improved estimation of data center resource utilization. *Future Gener. Comput. Syst.* **2020**, *104*, 212–224.
19. Youn, J.; Shim, J.; Lee, S.G. Efficient data stream clustering with sliding windows based on locality-sensitive hashing. *IEEE Access* **2018**, *6*, 63757–63776. [CrossRef]
20. Bahri, M.; Bifet, A.; Gama, J.; Gomes, H.M.; Maniu, S. Data stream analysis: Foundations, major tasks and tools. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, 1405. [CrossRef]
21. Baek, Y.; Yun, U.; Kim, H.; Nam, H.; Lee, G.; Yoon, E.; Vo, B.; Lin, J.C.W. Erasable pattern mining based on tree structures with damped window over data streams. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103735. [CrossRef]
22. Kim, J.; Yun, U.; Kim, H.; Ryu, T.; Lin, J.C.W.; Fournier-Vier, P.; Pedrycz, W. Average utility driven data analytics on damped windows for intelligent systems with data streams. *Int. J. Intell. Syst.* **2021**, *36*, 5741–5769. [CrossRef]
23. Zubaroğlu, A.; Atalay, V. Data stream clustering: A review. *Artif. Intell. Rev.* **2021**, *54*, 1201–1236. [CrossRef]
24. Tanbeer, S.K.; Ahmed, C.F.; Jeong, B.S.; Lee, Y.K. Sliding window-based frequent pattern mining over data streams. *Inf. Sci.* **2009**, *179*, 3843–3865. [CrossRef]

25. Giraud, C. *Introduction to High-Dimensional Statistics*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2021.
26. Assent, I. Clustering high dimensional data. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2*, 340–350. [CrossRef]
27. Peng, W.; Zhou, T.; Chen, Y. Enhancing mass spectrometry data analysis: A novel framework for calibration, outlier detection, and classification. *Pattern Recognit. Lett.* **2024**, *182*, 1–8. [CrossRef]
28. Harrou, F.; Bouyeddou, B.; Zerrouki, N.; Dairi, A.; Sun, Y.; Zerrouki, Y. Detecting the signs of desertification with Landsat imagery: A semi-supervised anomaly detection approach. *Results Eng.* **2024**, *22*, 102037. [CrossRef]
29. Tahvili, S.; Hatvani, L. Chapter three-transformation, vectorization, and optimization. In *Artificial Intelligence Methods for Optimization of the Software Testing Process, Ser. Uncertainty, Computational Techniques, and Decision Intelligence*; Academic Press: Cambridge, MA, USA, 2022; pp. 35–84.
30. Rozza, A.; Lombardi, G.; Ceruti, C.; Casiraghi, E.; Campadelli, P. Novel high intrinsic dimensionality estimators. *Mach. Learn.* **2012**, *89*, 37–65. [CrossRef]
31. Hawkins, D.M. *Identification of Outliers*; Chapman and Hall: London, UK, 1980; Volume 11.
32. Aggarwal, C.C. *Data Mining: The Textbook*; Springer: New York, NY, USA, 2015; Volume 1.
33. Smiti, A. A critical overview of outlier detection methods. *Comput. Sci. Rev.* **2020**, *38*, 100306. [CrossRef]
34. Škoda, P.; Adam, F. *Knowledge Discovery in Big Data from Astronomy and Earth Observation*; Elsevier: Amsterdam, The Netherlands, 2020.
35. Han, J.; Kamber, M.; Pei, J. (Eds.) Outlier Detection, The Morgan Kaufmann Series in Data Management Systems. In *Data Mining*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2012; pp. 543–584.
36. Gupta, M.; Gao, J.; Aggarwal, C.C.; Han, J. Outlier detection for temporal data: A survey. *IEEE Trans. Knowl. Data Eng.* **2013**, *26*, 2250–2267. [CrossRef]
37. Shi, Y.; Gong, J.; Deng, M.; Yang, X.; Xu, F. A graph-based approach for detecting spatial cross-outliers from two types of spatial point events. *Comput. Environ. Urban Syst.* **2018**, *72*, 88–103. [CrossRef]
38. Zheng, Y.; Zhang, H.; Yu, Y. Detecting collective anomalies from multiple spatio-temporal datasets across different domains. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 3–6 November 2015; ACM: New York, NY, USA, 2015; pp. 1–10.
39. Qin, S.J. Neural networks for intelligent sensors and control—Practical issues and some solutions. In *Neural Systems for Control*; Academic Press: Cambridge, MA, USA, 1997; pp. 213–234.
40. Han, J.; Pei, J.; Yin, Y.; Mao, R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.* **2004**, *8*, 53–87. [CrossRef]
41. Keogh, E.; Lonardi, S.; Chiu, B.Y.C. Finding surprising patterns in a time series database in linear time and space. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, AL, Canada, 23–26 July 2002; pp. 550–556.
42. Kern, R.; Al-Ubaidi, T.; Sabol, V.; Krebs, S.; Khodachenko, M.; Scherf, M. Astro-and Geoinformatics–Visually Guided Classification of Time Series Data. In *Knowledge Discovery in Big Data From Astronomy and Earth Observation*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 267–282.
43. Knapp, E.D.; Langill, J. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*; Syngress: Oxford, UK, 2014.
44. Kotu, V.; Deshpande, B. *Data Science: Concepts and Practice*; Morgan Kaufmann: Burlington, MA, USA, 2018.
45. Duraj, A.; Szczepaniak, P.S. Outlier detection in data streams—A comparative study of selected methods. *Procedia Comput. Sci.* **2021**, *192*, 2769–2778. [CrossRef]
46. Fernandes, G.; Rodrigues, J.J.; Carvalho, L.F.; Al-Muhtadi, J.F.; Proença, M.L. A comprehensive survey on network anomaly detection. *Telecommun. Syst.* **2019**, *70*, 447–489. [CrossRef]
47. Dwivedi, R.K.; Rai, A.K.; Kumar, R. Outlier detection in wireless sensor networks using machine learning techniques: A survey. In Proceedings of the 2020 International Conference on Electrical and Electronics Engineering (ICE3), Gorakhpur, India, 14–15 February 2020; IEEE: Piscataway, NY, USA, 2020; pp. 316–321.
48. Wang, H.; Bah, M.J.; Hammad, M. Progress in outlier detection techniques: A survey. *IEEE Access* **2019**, *7*, 107964–108000. [CrossRef]
49. Habeeb, R.A.A.; Nasaruddin, F.; Gani, A.; Hashem, I.A.T.; Ahmed, E.; Imran, M. Real-time big data processing for anomaly detection: A survey. *Int. J. Inf. Manag.* **2019**, *45*, 289–307. [CrossRef]
50. Samara, M.A.; Bennis, I.; Abouaissa, A.; Lorenz, P. A survey of outlier detection techniques in IoT: Review and classification. *J. Sens. Actuator Netw.* **2022**, *11*, 4. [CrossRef]
51. Gaddam, A.; Wilkin, T.; Angelova, M.; Gaddam, J. Detecting sensor faults, anomalies and outliers in the internet of things: A survey on the challenges and solutions. *Electronics* **2020**, *9*, 511. [CrossRef]
52. Souiden, I.; Omri, M.N.; Brahmi, Z. A survey of outlier detection in high dimensional data streams. *Comput. Sci. Rev.* **2022**, *44*, 100463. [CrossRef]
53. Molugaram, K.; Rao, G.S.; Shah, A.; Davergave, N. *Statistical Techniques for Transportation Engineering*; Butterworth-Heinemann: Portsmouth, NH, USA, 2017.
54. Ryu, M.; Lee, G.; Lee, K. Online sequential extreme studentized deviate tests for anomaly detection in streaming data with varying patterns. *Clust. Comput.* **2021**, *24*, 1975–1987. [CrossRef]

55. Leys, C.; Ley, C.; Klein, O.; Bernard, P.; Licata, L. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *J. Exp. Soc. Psychol.* **2013**, *49*, 764–766. [CrossRef]
56. Bhargavi, M.V.; Sireesha, V. A comparative study for statistical outlier detection using colon cancer data. *Adv. Appl. Stat.* **2022**, *72*, 41–54. [CrossRef]
57. Vieira, R.G.; Leone Filho, M.A.; Semolini, R. An Enhanced Seasonal-Hybrid ESD technique for robust anomaly detection on time series. *Symp. Bras. Redes Comput. Sist. Distrib.* **2018**, 281–294. [CrossRef]
58. Ray, S.; McEvoy, D.S.; Aaron, S.; Hickman, T.T.; Wright, A. Using statistical anomaly detection models to find clinical decision support malfunctions. *J. Am. Med. Inform. Assoc.* **2018**, *25*, 862–871. [CrossRef] [PubMed]
59. Saleem, S.; Aslam, M.; Shaukat, M.R. A Review and Empirical Comparison of univariate outlier Detection Methods. *Pak. J. Stat.* **2021**, *37*, 447–462.
60. Bhattacharya, S.; Beirlant, J. Outlier detection and a tail-adjusted boxplot based on extreme value theory. *arXiv* **2019**, arXiv:1912.02595.
61. Dai, W.; Mrkvička, T.; Sun, Y.; Genton, M.G. Functional outlier detection and taxonomy by sequential transformations. *Comput. Stat. Data Anal.* **2020**, *149*, 106960. [CrossRef]
62. Walker, M.L.; Dovoedo, Y.H.; Chakraborti, S.; Hilton, C.W. An improved boxplot for univariate data. *Am. Stat.* **2018**, *72*, 348–353. [CrossRef]
63. Rousseeuw, P.J.; Croux, C. Alternatives to the median absolute deviation. *J. Am. Stat. Assoc.* **1993**, *88*, 1273–1283. [CrossRef]
64. Devarakonda, N.; Subhani, S.; Basha, S.A.H. Outliers detection in regression analysis using partial least square approach. In Proceedings of the ICT and Critical Infrastructure: 48th Annual Convention of Computer Society of India, Visakhapatnam, India, 13–15 December 2013; Springer: Berlin/Heidelberg, Germany, 2014; Volume 2, pp. 125–135.
65. Sklar, A. Fonctions de répartition à n dimensions et leurs marges. *Publ. Inst. Statist. Univ. Paris* **1959**, *8*, 229–231.
66. Klein, N.; Kneib, T.; Marra, G.; Radice, R. Bayesian mixed binary-continuous copula regression with an application to childhood undernutrition. In *Flexible Bayesian Regression Modelling*; Academic Press: Cambridge, MA, USA, 2020; pp. 121–152.
67. Li, Z.; Zhao, Y.; Botta, N.; Ionescu, C.; Hu, X. COPOD: Copula-based outlier detection. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; IEEE: Piscataway, NY, USA, 2020; pp. 1118–1123.
68. Wang, Y.; Infield, D.G.; Stephen, B.; Galloway, S.J. Copula-based model for wind turbine power curve outlier rejection. *Wind Energy* **2014**, *17*, 1677–1688. [CrossRef]
69. Ghalem, S.K.; Kechar, B.; Bounceur, A.; Euler, R. A probabilistic multivariate copula-based technique for faulty node diagnosis in wireless sensor networks. *J. Netw. Comput. Appl.* **2019**, *127*, 9–25. [CrossRef]
70. Fang, G.; Pan, R. On multivariate copula modelling of dependent degradation processes. *Comput. Ind. Eng.* **2021**, *159*, 107450. [CrossRef]
71. Škorić, T.; Pantelić, D.; Jelenković, B.; Bajić, D. Noise reduction in two-photon laser scanned microscopic images by singular value decomposition with copula threshold. *Signal Process.* **2022**, *195*, 108486. [CrossRef]
72. Sheikhi, A.; Amirzadeh, V.; Mesiar, R. A comprehensive family of copulas to model bivariate random noise and perturbation. *Fuzzy Sets Syst.* **2021**, *415*, 27–36. [CrossRef]
73. Wang, M.L.; Lynch, J.P.; Sohn, H. (Eds.) Sensing hardware and data collection methods. In *Sensor Technologies for Civil Infrastructures*; Woodhead Publishing: Sawston, UK, 2014; Volume 1.
74. Carson, E.; Cobelli, C. *Modelling Methodology for Physiology and Medicine*, 2nd ed.; Newnes: London, UK, 2013.
75. Theodoridis, S. Bayesian learning: Inference and the EM algorithm. In *Machine Learning*; Academic Press: Cambridge, MA, USA, 2020; pp. 595–646.
76. Haldar, S.K. Statistical and geostatistical applications in geology. In *Mineral Exploration*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 167–194.
77. Goldstein, M.; Dengel, A. Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. In Proceedings of the 35th German Conference on Artificial Intelligence KI-2012, Saarbrücken, Germany, 24–27 September 2012; Volume 1, pp. 59–63.
78. Latecki, L.J.; Lazarevic, A.; Pokrajac, D. Outlier detection with kernel density functions. In Proceedings of the International Workshop on Machine Learning and Data Mining in Pattern Recognition, New York, NY, USA, 15–20 July 2017; Springer: Berlin/Heidelberg, Germany, 2007; pp. 61–75.
79. Schubert, E.; Zimek, A.; Kriegel, H.P. Generalized outlier detection with flexible kernel density estimates. In Proceedings of the 2014 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics 2014, Philadelphia, PA, USA, 24–26 April 2014; pp. 542–550.
80. Abdulghafoor, S.A.; Mohamed, L.A. A local density-based outlier detection method for high dimension data. *Int. J. Nonlinear Anal. Appl.* **2022**, *13*, 1683–1699.
81. Ghoting, A.; Parthasarathy, S.; Otey, M.E. Fast mining of distance-based outliers in high-dimensional datasets. *Data Min. Knowl. Discov.* **2008**, *16*, 349–364. [CrossRef]
82. Vu, N.H.; Gopalkrishnan, V. Efficient pruning schemes for distance-based outlier detection. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Bled, Slovenia, 6–10 September 2009; Springer: Berlin/Heidelberg, Germany, 2009; pp. 160–175.

83. Navarro, J.; de Diego, I.M.; Fernández, R.R.; Moguerza, J.M. Triangle-based outlier detection. *Pattern Recognit. Lett.* **2022**, *156*, 152–159. [CrossRef]
84. Angiulli, F.; Fassetti, F. Uncertain distance-based outlier detection with arbitrarily shaped data objects. *J. Intell. Inf. Syst.* **2021**, *57*, 1–24. [CrossRef]
85. Román, I.S.; de Diego, I.M.; Conde, C.; Cabello, E. Outlier trajectory detection through a context-aware distance. *Pattern Anal. Appl.* **2019**, *22*, 831–839. [CrossRef]
86. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000; ACM: New York, NY, USA, 2000; pp. 93–104.
87. Bai, M.; Wang, X.; Xin, J.; Wang, G. An efficient algorithm for distributed density-based outlier detection on big data. *Neurocomputing* **2016**, *181*, 19–28. [CrossRef]
88. Zhang, L.; Lin, J.; Karim, R. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowl. Based Syst.* **2018**, *139*, 50–63. [CrossRef]
89. Pokrajac, D.; Lazarevic, A.; Latecki, L.J. Incremental local outlier detection for data streams. In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining, Honolulu, HI, USA, 1 March–5 April 2007; IEEE: Piscataway, NY, USA, 2007; pp. 504–515.
90. Degirmenci, A.; Karal, O. iMCOD: Incremental multi-class outlier detection model in data streams. *Knowl. Based Syst.* **2022**, *258*, 109950. [CrossRef]
91. 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]
92. Yan, Y.; Cao, L.; Kulhman, C.; Rundensteiner, E. Distributed local outlier detection in big data. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2017, Halifax, NS, Canada, 13–17 August 2017; pp. 1225–1234.
93. Chen, L.; Wang, W.; Yang, Y. CELOF: Effective and fast memory efficient local outlier detection in high-dimensional data streams. *Appl. Soft Comput.* **2021**, *102*, 107079. [CrossRef]
94. Cassisi, C.; Ferro, A.; Giugno, R.; Pigola, G.; Pulvirenti, A. Enhancing density-based clustering: Parameter reduction and outlier detection. *Inf. Syst.* **2013**, *38*, 317–330. [CrossRef]
95. Nozad, S.A.N.; Haeri, M.A.; Folino, G. SDCOR: Scalable density-based clustering for local outlier detection in massive-scale datasets. *Knowl. Based Syst.* **2021**, *228*, 107256. [CrossRef]
96. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. Density-based spatial clustering of applications with noise. In Proceedings of the International Conferences Knowledge Discovery and Data Mining 1996, Portland, OR, USA, 2–4 August 1996; p. 240.
97. Degirmenci, A.; Karal, O. Efficient density and cluster based incremental outlier detection in data streams. *Inf. Sci.* **2022**, *607*, 901–920. [CrossRef]
98. Kriegel, H.P.; Schubert, M.; Zimek, A. Angle-based outlier detection in high-dimensional data. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2008, Las Vegas, NV, USA, 24–27 August 2008; pp. 444–452.
99. Al-taei, R.; Haeri, M.A. An ensemble angle-based outlier detection for big data. In Proceedings of the International Congress on High-Performance Computing and Big Data Analysis, Tehran, Iran, 23–25 April 2019; Springer: Cham, Switzerland, 2019; pp. 98–108.
100. Ye, H.; Kitagawa, H.; Xiao, J. Continuous angle-based outlier detection on high-dimensional data streams. In Proceedings of the 19th International Database Engineering & Applications Symposium, Yokohama, Japan, 13–15 July 2015; pp. 162–167.
101. Thordsen, E.; Schubert, E. ABID: Angle based intrinsic dimensionality. In Proceedings of the International Conference on Similarity Search and Applications, Copenhagen, Denmark, 30 September–2 October 2020; Springer: Cham, Switzerland, 2020; pp. 218–232.
102. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
103. Urso, A.; Fiannaca, A.; La Rosa, M.; Ravì, V.; Rizzo, R. Data mining: Classification and prediction. *Encycl. Bioinform. Comput. Biol. ABC Bioinform.* **2018**, *1*, 384.
104. Singh, P.K.; Gupta, S.; Vashistha, R.; Nandi, S.K.; Nandi, S. Machine learning based approach to detect position falsification attack in VANETs. In Proceedings of the Security and Privacy: 2nd ISEA International Conference, ISEA-ISAP 2018, Jaipur, India, 9–11 January 2019; Springer: Singapore, 2019; pp. 166–178.
105. Parras, J.; Zazo, S. Using one class SVM to counter intelligent attacks against an SPRT defense mechanism. *Ad Hoc Netw.* **2019**, *94*, 101946. [CrossRef]
106. Sumathy, S.; Revathy, M.; Manikandan, R. Improving the state of materials in cybersecurity attack detection in 5G wireless systems using machine learning. *Mater. Today Proc.* **2023**, *81*, 700–707. [CrossRef]
107. Erfani, S.M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognit.* **2016**, *58*, 121–134. [CrossRef]
108. Zhou, X.; Zhang, X.; Wang, B. Online support vector machine: A survey. In Proceedings of the Harmony Search Algorithm 2nd International Conference on Harmony Search Algorithm (ICHSA2015), Seoul, Republic of Korea, 19–21 August 2015; Springer: Berlin/Heidelberg, Germany, 2016; pp. 269–278.

109. Martín, L.; Sánchez, L.; Lanza, J.; Sotres, P. Development and evaluation of Artificial Intelligence techniques for IoT data quality assessment and curation. *Internet Things* **2023**, *22*, 100779. [CrossRef]
110. Rosenblatt, F. *The Perceptron, a Perceiving and Recognizing Automaton (Project PARA)*; Cornell Aeronautical Laboratory: Buffalo, NY, USA, 1957.
111. Krishnan, S. Machine learning for biomedical signal analysis. In *Biomedical Signal Analysis for Connected Healthcare*; Elsevier: Amsterdam, The Netherlands, 2021; pp. 223–264.
112. Al-Jabery, K.; Obafemi-Ajayi, T.; Olbricht, G.; Wunsch, D. *Computational Learning Approaches to Data Analytics in Biomedical Applications*; Academic Press: Cambridge, MA, USA, 2019.
113. Svozil, D.; Kvasnicka, V.; Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **1997**, *39*, 43–62. [CrossRef]
114. Iqbal, A.; Aftab, S. A feed-forward and pattern recognition ANN model for network intrusion detection. *Int. J. Comput. Netw. Inf. Secur.* **2019**, *11*, 19. [CrossRef]
115. Ullah, I.; Mahmoud, Q.H. An anomaly detection model for IoT networks based on flow and flag features using a feed-forward neural network. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; IEEE: Piscataway, NY, USA, 2022; pp. 363–368.
116. Li, H.; Wang, X.; Yang, Z.; Ali, S.; Tong, N.; Baseer, S. Correlation-Based Anomaly Detection Method for Multi-sensor System. *Comput. Intell. Neurosci.* **2022**, *2022*, 4756480. [CrossRef]
117. Kang, Z.; Yang, B.; Nielsen, M.; Deng, L.; Yang, S. A buffered online transfer learning algorithm with multi-layer network. *Neurocomputing* **2022**, *488*, 581–597. [CrossRef]
118. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
119. DiPietro, R.; Hager, G.D. Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*; Academic Press: Cambridge, MA, USA, 2020; pp. 503–519.
120. Singh, E.; Kuzhagaliyeva, N.; Sarathy, S.M. Using deep learning to diagnose preignition in turbocharged spark-ignited engines. In *Artificial Intelligence and Data Driven Optimization of Internal Combustion Engines*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 213–237.
121. Gupta, T.K.; Raza, K. Optimization of ANN architecture: A review on nature-inspired techniques. In *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*; Academic Press: Cambridge, MA, USA, 2019; pp. 159–182.
122. Zhu, R.; Tu, X.; Huang, J.X. Deep learning on information retrieval and its applications. In *Deep Learning for Data Analytics*; Academic Press: Cambridge, MA, USA, 2020; pp. 125–153.
123. Muharemi, F.; Logofătu, D.; Leon, F. Machine learning approaches for anomaly detection of water quality on a real-world data set. *J. Inf. Telecommun.* **2019**, *3*, 294–307. [CrossRef]
124. Ackerson, J.M.; Dave, R.; Seliya, N. Applications of recurrent neural network for biometric authentication & anomaly detection. *Information* **2021**, *12*, 272. [CrossRef]
125. Jeong, S.; Ferguson, M.; Law, K.H. Sensor data reconstruction and anomaly detection using bidirectional recurrent neural network. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2019, Denver, CO, USA, 3–7 March 2019; Volume 10970, pp. 157–167.
126. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
127. Ankit, U. Transformer Neural Network: Step-By-Step Breakdown of the Beast. 2020. Available online: https://towardsdatascience.com/transformer-neural-network-step-by-step-breakdown-of-the-beast-b3e096dc857f (accessed on 20 September 2023).
128. Al Mamun, S.A.; Beyaz, M. LSTM Recurrent Neural Network (RNN) for Anomaly Detection in Cellular Mobile Networks. In Proceedings of the Machine Learning for Networking: First International Conference MLN 2018, Paris, France, 27–29 November 2018; pp. 222–237.
129. Muhuri, P.S.; Chatterjee, P.; Yuan, X.; Roy, K.; Esterline, A. Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. *Information* **2020**, *11*, 243. [CrossRef]
130. Sagheer, A.; Hamdoun, H.; Youness, H. Deep LSTM-based transfer learning approach for coherent forecasts in hierarchical time series. *Sensors* **2021**, *21*, 4379. [CrossRef] [PubMed]
131. Bleiweiss, A. LSTM Neural Networks for Transfer Learning in Online Moderation of Abuse Context. In Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019), Prague, Czech Republic, 19–21 February 2019; pp. 112–122.
132. Negi, N.; Jelassi, O.; Chaouchi, H.; Clemençon, S. Distributed online Data Anomaly Detection for connected vehicles. In Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 19–21 February 2020; IEEE: Piscataway, NY, USA, 2020; pp. 616–621.
133. Raj, P.; Evangeline, P. *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*; Academic Press: Cambridge, MA, USA, 2020.
134. Pavithra, V.; Jayalakshmi, V. Smart energy and electric power system: Current trends and new intelligent perspectives and introduction to AI and power system. In *Smart Energy and Electric Power Systems*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 19–36.
135. Hung, C.L. Deep learning in biomedical informatics. In *Intelligent Nanotechnology*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 307–329.

136. Teuwen, J.; Moriakov, N. Convolutional neural networks. In *Handbook of Medical Image Computing and Computer Assisted Intervention*; Academic Press: Cambridge, MA, USA, 2020; pp. 481–501.
137. Jeon, W.; Ko, G.; Lee, J.; Lee, H.; Ha, D.; Ro, W.W. Deep learning with GPUs. *Adv. Comput.* **2021**, *122*, 167–215.
138. Mishra, S.; Tripathy, H.K.; Mallick, P.K.; Sangaiah, A.K.; Chae, G.S. (Eds.) *Cognitive Big Data Intelligence with a Metaheuristic Approach*; Academic Press: Cambridge, MA, USA, 2021.
139. Hinton, G.E.; Osindero, S.; Teh, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
140. Mocanu, E.; Nguyen, P.H.; Gibescu, M. Deep learning for power system data analysis. In *Big Data Application in Power Systems*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 125–158.
141. Liu, H. *Wind Forecasting in Railway Engineering*; Elsevier: Amsterdam, The Netherlands, 2021.
142. Talapula, D.K.; Kumar, A.; Ravulakollu, K.K.; Kumar, M. Anomaly Detection in Online Data Streams Using Deep Belief Neural Networks. In Proceedings of the Doctoral Symposium on Computational Intelligence, Lucknow, India, 3 March 2023; Springer: Singapore, 2023; pp. 729–749.
143. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems NIPS'14, Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 2672–2680.
144. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
145. Aggarwal, A.; Mittal, M.; Battineni, G. Generative adversarial network: An overview of theory and applications. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100004. [CrossRef]
146. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems NeurIPS 2017, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
147. Li, G.; Duan, Z.; Liang, L.; Zhu, H.; Hu, A.; Cui, Q.; Chen, B.; Hu, W. Outlier data mining method considering the output distribution characteristics for photovoltaic arrays and its application. *Energy Rep.* **2020**, *6*, 2345–2357. [CrossRef]
148. Srinu, S.; Mishra, A.K. Efficient elimination of erroneous nodes in cooperative sensing for cognitive radio networks. *Comput. Electr. Eng.* **2016**, *52*, 284–292. [CrossRef]
149. Zhao, Y.; Lehman, B.; Ball, R.; Mosesian, J.; de Palma, J.F. Outlier detection rules for fault detection in solar photovoltaic arrays. In Proceedings of the 2013 28th Annual IEEE Applied Power Electronics Conference and Exposition (APEC), Long Beach, CA, USA, 17–21 March 2013; IEEE: Piscataway, NY, USA, 2013; pp. 2913–2920.
150. Schlechtingen, M.; Santos, I.F. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mech. Syst. Signal Process.* **2011**, *25*, 1849–1875. [CrossRef]
151. 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.; 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]
152. Owolabi, O.; Okoh, D.; Rabiu, B.; Obafaye, A.; Dauda, K. A median absolute deviation-neural network (MAD-NN) method for atmospheric temperature data cleaning. *MethodsX* **2021**, *8*, 101533. [CrossRef] [PubMed]
153. Bae, I.; Ji, U. Application of Outlier Detection and Smoothing Algorithm for Monitoring Water Level and Discharge by Ultrasonic Sensor. In Proceedings of the AGU Fall Meeting Abstracts, San Francisco, CA, USA, 9–13 December 2019; Volume 2019, p. H53K-1913.
154. Belkhouche, F. Robust calibration of MEMS accelerometers in the presence of outliers. *IEEE Sens. J.* **2022**, *22*, 9500–9508. [CrossRef]
155. Diaz-Rozo, J.; Bielza, C.; Larrañaga, P. Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes. *IEEE Internet Things J.* **2018**, *5*, 3533–3547. [CrossRef]
156. Reddy, A.; Ordway-West, M.; Lee, M.; Dugan, M.; Whitney, J.; Kahana, R.; Ford, B.; Muedsam, J.; Henslee, A.; Rao, M. Using gaussian mixture models to detect outliers in seasonal univariate network traffic. In Proceedings of the 2017 IEEE Security and Privacy Workshops (SPW), San Jose, CA, USA, 25 May 2017; IEEE: Piscataway, NY, USA, 2017; pp. 229–234.
157. Kalaycı, İ.; Ercan, T. Anomaly detection in wireless sensor networks data by using histogram based outlier score method. In Proceedings of the 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 19–21 October 2018; IEEE: Piscataway, NY, USA, 2018; pp. 1–6.
158. Çakmakçı, S.D.; Kemmerich, T.; Ahmed, T.; Baykal, N. Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm. *J. Netw. Comput. Appl.* **2020**, *168*, 102756. [CrossRef]
159. Saeed, M.M. A real-time adaptive network intrusion detection for streaming data: A hybrid approach. *Neural Comput. Appl.* **2022**, *34*, 6227–6240. [CrossRef]
160. Alamaniotis, M. Fuzzy Integration of kernel-based Gaussian Processes applied to Anomaly Detection in Nuclear Security. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), Chania Crete, Greece, 12–14 July 2021*; IEEE: Piscataway, NY, USA, 2021; pp. 1–4.
161. Bhattacharjee, S.; Marchang, N. Malicious user detection with local outlier factor during spectrum sensing in cognitive radio network. *Int. J. Ad Hoc Ubiquitous Comput.* **2019**, *30*, 215–223. [CrossRef]
162. Chhetry, B.; Marchang, N. Detection of primary user emulation attack (PUEA) in cognitive radio networks using one-class classification. *arXiv* **2021**, arXiv:2106.10964.

163. Baek, S.; Kwon, D.; Suh, S.C.; Kim, H.; Kim, I.; Kim, J. Clustering-based label estimation for network anomaly detection. *Digit. Commun. Netw.* **2021**, *7*, 37–44. [CrossRef]
164. Premkumar, M.; Ashokkumar, S.R.; Jeevanantham, V.; Mohanbabu, G.; AnuPallavi, S. Scalable and energy efficient cluster based anomaly detection against denial of service attacks in wireless sensor networks. *Wirel. Pers. Commun.* **2023**, *129*, 2669–2691. [CrossRef]
165. Yang, L.; Lu, Y.; Yang, S.X.; Guo, T.; Liang, Z. A secure clustering protocol with fuzzy trust evaluation and outlier detection for industrial wireless sensor networks. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4837–4847. [CrossRef]
166. Jha, H.S.; Khanal, A.; Seikh, H.M.D.; Lee, W.J. A comparative study on outlier detection techniques for noisy production data from unconventional shale reservoirs. *J. Nat. Gas Sci. Eng.* **2022**, *105*, 104720. [CrossRef]
167. Soumya, T.R.; Revathy, S. A Novel Approach for Cyber Threat Detection Based on Angle-Based Subspace Anomaly Detection. *Cybern. Syst.* **2022**, 1–10. [CrossRef]
168. Vanitha, N.; Ganapathi, P. Traffic analysis of UAV networks using enhanced deep feed forward neural networks (EDFFNN). In *Handbook of Research on Machine and Deep Learning Applications for Cyber Security*; IGI Global: Hershey, PA, USA, 2020; pp. 219–244.
169. Reddy, D.K.; Behera, H.S.; Nayak, J.; Vijayakumar, P.; Naik, B.; Singh, P.K. Deep neural network based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, 4121. [CrossRef]
170. 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]
171. Adkisson, M.; Kimmell, J.C.; Gupta, M.; Abdelsalam, M. Autoencoder-based anomaly detection in smart farming ecosystem. In Proceedings of the 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 15–18 December 2021; IEEE: Piscataway, NY, USA, 2021; pp. 3390–3399.
172. Han, P.; Ellefsen, A.L.; Li, G.; Holmeset, F.T.; Zhang, H. Fault detection with LSTM-based variational autoencoder for maritime components. *IEEE Sens. J.* **2021**, *21*, 21903–21912. [CrossRef]
173. Alabadi, M.; Celik, Y. Detection for cyber-security based on convolution neural network: A survey. In Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 26–28 June 2020; IEEE: Piscataway, NY, USA, 2020; pp. 1–14.
174. Sun, H.; Chen, M.; Weng, J.; Liu, Z.; Geng, G. Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10880–10893. [CrossRef]
175. Tschuchnig, M.E.; Gadermayr, M. Anomaly detection in medical imaging-a mini review. In *Data Science–Analytics and Applications: Proceedings of the 4th International Data Science Conference—iDSC2021, Online, 16–18 October 2021*; Springer: Wiesbaden, Germany, 2022; pp. 33–38.
176. Arabahmadi, M.; Farahbakhsh, R.; Rezazadeh, J. Deep learning for smart Healthcare—A survey on brain tumor detection from medical imaging. *Sensors* **2022**, *22*, 1960. [CrossRef]
177. Qiao, Y.; Cui, X.; Jin, P.; Zhang, W. Fast outlier detection for high-dimensional data of wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720963835. [CrossRef]
178. Sarkar, N.; Keserwani, P.K.; Govil, M.C. A better and fast cloud intrusion detection system using improved squirrel search algorithm and modified deep belief network. *Clust. Comput.* **2023**, *27*, 1699–1718. [CrossRef]
179. Deecke, L.; Vandermeulen, R.; Ruff, L.; Mandt, S.; Kloft, M. Image anomaly detection with generative adversarial networks. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference ECML PKDD 2018, Dublin, Ireland, 10–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–17.
180. Jiang, T.; Li, Y.; Xie, W.; Du, Q. Discriminative reconstruction constrained generative adversarial network for hyperspectral anomaly detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4666–4679. [CrossRef]
181. Jin, P.; Mou, L.; Xia, G.S.; Zhu, X.X. Anomaly detection in aerial videos with transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5628213. [CrossRef]
182. Chen, Z.; Chen, D.; Zhang, X.; Yuan, Z.; Cheng, X. Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. *IEEE Internet Things J.* **2021**, *9*, 9179–9189. [CrossRef]
183. Zhang, S.; Liu, Y.; Zhang, X.; Cheng, W.; Chen, H.; Xiong, H. Cat: Beyond efficient transformer for content-aware anomaly detection in event sequences. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 4541–4550.
184. Zhang, J.; Zhao, H.; Li, J. TRS: Transformers for remote sensing scene classification. *Remote Sens.* **2021**, *13*, 4143. [CrossRef]
185. ODDS. Outliers Detection Datasets. Available online: https://odds.cs.stonybrook.edu/ (accessed on 23 July 2024).
186. IEEE Dataport. IEEE Dataport Datasets. Available online: https://ieee-dataport.org/datasets (accessed on 23 July 2024).
187. University of California Irving. University of California Irving Database. Available online: https://kdd.ics.uci.edu/databases/ (accessed on 23 July 2024).
188. UCI Machine Learning Repository. KDD Cup 1999 Data. University of California, Irvine. 1999. Available online: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 29 July 2024).
189. University of New Brunswick. NSL-KDD Dataset. *Information Security Centre of Excellence, University of New Brunswick.* 2009. Available online: http://www.unb.ca/cic/datasets/nsl.html (accessed on 29 July 2024).
190. Koppula, M.; Joseph, L. A Real Time Dataset IDSIoT 2024. *IEEE Data Port.* 2024. [CrossRef]

191. Pack, M.L. Corel Histogram Dataset. Available online: https://www.mlpack.org/datasets/ (accessed on 23 July 2024).
192. Pahl, M.O.; Aubet, F.X. All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018; IEEE: Piscataway, NY, USA, 2018; pp. 72–80.
193. Intel Berkeley Research Lab. Intel Berkeley Research Lab Sensor Data. *Intel Corporation.* 2004. Available online: http://db.csail.mit.edu/labdata/labdata.html (accessed on 29 July 2024).
194. Zhang, Y.; Zhu, Y.; Nichols, E.; Wang, Q.; Zhang, S.; Smith, C.; Howard, S. A Poisson-Gaussian Denoising Dataset with Real Fluorescence Microscopy Images. *arXiv* **2018**, arXiv:1812.10366.
195. Jorge, R.-O.; Davide, A.; Alessandro, G.; Luca, O.; Xavier, P. *Human Activity Recognition Using Smartphones*; UCI Machine Learning Repository; University of California: Irvine, CA, USA, 2012. [CrossRef]
196. Billur, B.; Kerem, A. *Daily and Sports Activities*; UCI Machine Learning Repository; University of California: Irvine, CA, USA, 2013. [CrossRef]
197. Center for Atmospheric Research. Tropospheric Data Acquisition Network (TRODAN) Data. 2013. Available online: https://carnasrda.com/trodan_data (accessed on 29 July 2024).
198. Canadian Institute for Cybersecurity. CICIDS2017 Dataset. *University of New Brunswick.* 2017. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 29 July 2024).
199. University of New South Wales. BoT-IoT Dataset. *UNSW Canberra Cyber.* 2018. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed on 29 July 2024).
200. University of California. Merced. UC Merced Land Use Dataset. 2010. Available online: http://weegee.vision.ucmerced.edu/datasets/landuse.html (accessed on 29 July 2024).
201. Xia, G.-S. AID: Aerial Image Dataset. Wuhan University. 2017. Available online: https://captain-whu.github.io/DiRS/ (accessed on 29 July 2024).
202. Haikel, H. *NWPU-RESISC45 Dataset with 12 Classes*; Figshare: London, UK, 2021. [CrossRef]
203. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1155–1167. [CrossRef]