minerals

MDPI

Article

Effective Outlier Detection for Ensuring Data Quality in Flotation Data Modelling Using Machine Learning (ML) Algorithms

Clement Lartey [1,2,*], Jixue Liu [2], Richmond K. Asamoah [1], Christopher Greet [3], Massimiliano Zanin [1,4] and William Skinner [1]

[1] Future Industries Institute, University of South Australia, Adelaide, SA 5095, Australia
[2] UniSA STEM, University of South Australia, Adelaide, SA 5095, Australia
[3] Magotteaux Australia Pty. Ltd., Wingfield, Adelaide, SA 5013, Australia
[4] School of Chemical Engineering, The University of Adelaide, Adelaide, SA 5005, Australia
[*] Correspondence: clement.lartey@mymail.unisa.edu.au

Abstract: Froth flotation, a widely used mineral beneficiation technique, generates substantial volumes of data, offering the opportunity to extract valuable insights from these data for production line analysis. The quality of flotation data is critical to designing accurate prediction models and process optimisation. Unfortunately, industrial flotation data are often compromised by quality issues such as outliers that can produce misleading or erroneous analytical results. A general approach is to preprocess the data by replacing or imputing outliers with data values that have no connection with the real state of the process. However, this does not resolve the effect of outliers, especially those that deviate from normal trends. Outliers often occur across multiple variables, and their values may occur in normal observation ranges, making their detection challenging. An unresolved challenge in outlier detection is determining how far an observation must be to be considered an outlier. Existing methods rely on domain experts' knowledge, which is difficult to apply when experts encounter large volumes of data with complex relationships. In this paper, we propose an approach to conduct outlier analysis on a flotation dataset and examine the efficacy of multiple machine learning (ML) algorithms—including k-Nearest Neighbour (kNN), Local Outlier Factor (LOF), and Isolation Forest (ISF)—in relation to the statistical 2σ rule for identifying outliers. We introduce the concept of "quasi-outliers" determined by the 2σ threshold as a benchmark for assessing the ML algorithms' performance. The study also analyses the mutual coverage between quasi-outliers and outliers from the ML algorithms to identify the most effective outlier detection algorithm. We found that the outliers by kNN cover outliers of other methods. We use the experimental results to show that outliers affect model prediction accuracy, and excluding outliers from training data can reduce the average prediction errors.

Keywords: froth flotation; outlier detection; machine learning; prediction error; data quality

check for updates

Citation: Lartey, C.; Liu, J.; Asamoah, R.K.; Greet, C.; Zanin, M.; Skinner, W. Effective Outlier Detection for Ensuring Data Quality in Flotation Data Modelling Using Machine Learning (ML) Algorithms. *Minerals* **2024**, *14*, 925. https://doi.org/10.3390/min14090925

Academic Editors: Miguel Maldonado and Luis Vinnett

Received: 30 July 2024
Revised: 5 September 2024
Accepted: 6 September 2024
Published: 10 September 2024

1. Introduction

Froth flotation is a physicochemical separation of economically valuable minerals of interest from their gangue [1,2]. This separation process occurs in organised cells in which the feed material (i.e., ore) is treated until the valuable minerals are sufficiently recovered. In most industrial operations, sensors are used to measure key parameters of the flotation process, leading to the production of large volumes of data for analysis. Recent advances in machine learning (ML) application offer opportunities to effectively use flotation data to design predictive and process control models for process optimisation. However, sensed flotation data are prone to quality issues, mainly outliers that compromise the reliability of the data and the accuracy of models derived from them. To leverage valuable insights from flotation data analytics, it is critical to have high-quality data to enable ML models to learn meaningful relationships to effectively monitor control systems, improve performance, and optimise processes.

Enhancing data quality is necessary, as outliers can interfere with experimental analysis leading to biased predictions, misleading insights, and reduced generalisation [3]. Outliers may not always be bad observations in the dataset. It is worth mentioning that outliers can have exceptional information, in which case further investigation may be needed to ascertain their inclusion or removal from the dataset. As such, researchers scrutinise outliers to understand the factors that contributed to their generation or unique circumstances that might have influenced their existence. This has facilitated the application of outlier detection across several domains, including fraud detection [4], network intrusion [5], disease diagnosis [6], and fault detection [7]. Despite its acknowledged significance in diverse fields, outlier detection has not received adequate attention in mineral processing data analytics, representing a relatively under-explored topic. This limited focus can be attributed to (1) outliers often perceived as errors to be discarded rather than interesting behaviours worth investigating, (2) the inherent complexity of data, which makes it challenging to accurately identify outliers, and (3) the lack of domain-specific methods for the identification and interpretation of outliers.

Outliers are observations that deviate from a body of data [8,9]. They can generally be classified into three main categories, namely point outliers, collective outliers, and contextual outliers. Point outliers refer to observations that deviate extremely from the overall distribution of a dataset [10]. Collective outliers describe a group of observations that deviate from the distribution of the dataset [11]. Contextual outliers refer to observations that are extremely different in a specific context [9,12]. For example, a summer temperature of 30 °C is normal but likely to be an outlier when recorded during winter. Within the mineral processing industry, factors such as faulty sensors, equipment malfunction, improper handling of missing data values, and unexpected fluctuations can produce any of these types of outliers in the production data [13,14]. As such, outliers should be carefully investigated using appropriate methods to effectively monitor process equipment and the data they generate. More importantly, outliers should be properly managed before making decisions based on analysis of the production data.

The flotation data represent dynamic relationships of key variables including feed variables (feed mineralogy, particle size, throughput, liberation), hydrodynamic variables (bubble size, air flow rate, froth depth, impeller speed), and chemical variables (reagent dosages, pulp chemical conditions). The interdependence of these variables makes it arduous to justify an observation as an outlier within the intricate web of relationships it shares with other variables. For instance, a decrease in Eh values in a flotation pulp measurement may not be an outlier observation. Instead, it may be attributable to an elevated iron sulphide content in the feed [15]. In addition, during comminution, changes that occur in mineralogy and grinding media can impact significant changes in the pulp chemistry of flotation feed [16,17]. Again, these changes may not be outliers. Furthermore, sensors used in harsh mineral processing environments may experience a breakdowns or failures, yet they may continue to record data from the operation, leading to compromised and potentially inaccurate readings [18]. Such variable associations and equipment conditions complicate the distinction between instabilities and outliers in the flotation data. To enhance the quality of flotation data, methods for outlier detection should be critically explored while considering the intricate relationships among multiple variables.

Studies on outlier detection spans several decades and can be broadly categorised as (1) statistical-based, (2) distance-based, (3) density-based, and (4) prediction-based techniques [19]. Statistical methods such as Grubb's test [20], Doerffel's test [21], Dixon's test [3], Peirce's test [22] and Chauvenet's test [23] are well known and efficient in detecting point outliers, especially those that occur in univariate datasets. Other works [24–26] have reported robust statistical methods of assessing outliers.

In recent years, a boxplot [27] technique for outlier detection has gained popularity in engineering domains. The boxplot utilises a concept of interquartile range (IQR) to visualise outliers. The IQR is computed as $IQR = Q3 - Q1$, where $Q1$ is the first quartile and $Q3$ is the third quartile such that observations beyond the range $Q1 - 1.5(IQR)$ to $Q3 + 1.5(IQR)$

are considered potential outliers [28]. Other studies have used the minimum covariance determinant (MCD) and the minimum volume ellipsoid (MVE) to analyse multivariate data for outliers [12]. However, both MCD and MVE have some limitations as they become ineffective if the data dimension increases. Although statistically-based methods are easy to implement, they are mostly sensitive to outliers, as their computation relies on the properties of the mean, median, and standard deviation of the data. In addition, their concept follows an underlying assumption of normally distributed data, which is often not the case in real-world data. Furthermore, they are ineffective in detecting multivariate outliers, especially those occurring in high-dimension datasets.

Alternatively, distance-based methods [29,30] offer solutions to mitigate the limitations of statistical methods. Distance-based methods use distance metrics such as the Euclidean distance to calculate the distance between observations and identify outliers based on these distance relationships. Knorr and Ng [29] proposed a classical distance-based outlier detection technique. They defined a unified notion of outliers as follows. An object O in a dataset T is a *UO(p, D)-outlier* if at least fraction p of the objects in T are $\geq$distance D from O [29]. Ramaswamy et al. [31] improved this concept by computing the distance from the k-Nearest Neighbour (kNN) of observations and considered potential outliers as observations that fell beyond a specified neighbourhood. Distance-based methods have several drawbacks, including (1) the assumption that data are uniformly distributed, which may not hold for heterogeneous data with varying distributions, (2) algorithm complexities which arise with high-dimension datasets, and (3) an ineffective detection of outliers existing within dense cluster regions.

To overcome the shortfalls of distance-based methods, researchers have explored density-based outlier detection methods [32,33]. The most widely used density-based method is the Local Outlier Factor (LOF) [34]. It adopts the concept of comparing the local density of an observation to the density of its neighbours. An observation is considered an outlier if it lies in a lower-density region compared to the local density of its neighbours. A score is computed to describe the degree of '*outlierness*'. This score is used to identify the exceptions in the dataset whose divergence is not easily detected as well as those that exist in high-dimensional subspaces [35,36]. Recently, several variants of the LOF have been explored, including Local Outlier Probability (LoOP) [37], Local Correlation Integral (LOCI) [38], Local Sparsity Coefficient (LSC) [39], and Local Distance-based Outlier Factor (LDOF) [40]. Although density-based methods can capture local outliers, they tend to be ineffective when low-density patterns occur in a given dataset [41,42].

The task of detecting and confirming outliers in the flotation data is not straightforward given the complexities associated with multiple variables as well as the diverse principles underlying various detection methods. Individual methods are effective only if their principles of detection apply. This means different methods would detect different outliers. As such, it is unclear what method to use and what threshold to set.

In this research, we propose an approach to conduct outlier detection in flotation data, addressing two main challenges in complex industrial processes: (1) The first is the presence of atypical data points that fall within the range of normal observations but represent anomalous process conditions. These points, while numerically similar to normal data, may indicate subtle deviations in the flotation process that are important to identify. and (2) The second is the multidimensional nature of outliers in flotation data, where observations may appear normal when viewed from one perspective (or in one dimension) but exhibit anomalous behaviour when considered in the context of other variables. Our approach consists of four parts. First, a standard deviation factor of the outlier scores is used to determine which observations in the data are outliers. Second, we use a naive algorithm called *trend differential* to identify quasi-outliers, including observations that visually form sharp peaks on the input features. Thirdly, we use different machine learning (ML) algorithms to identify outliers in the dataset from different perspectives. Fourthly, we analyse the coverage of quasi-outliers by outliers from the ML algorithms to confirm valid outliers and determine the effectiveness of the ML algorithms. The ML algorithms used in our work include k-Nearest Neighbour (kNN), Local Outlier Factor (LOF), and Isolation Forest (ISF).

Our approach addresses two key questions: (1) should multiple methods be used in detecting outliers and (2) how should the methods and their results be compared?

The contributions of this study are as follows:

1. The standard deviation factor of two (2) is verified to be a suitable value to define the threshold for outlier detection.
2. A method called trend differential is proposed to systematically identify visual outliers called quasi-outliers. These outliers are important as a starting point for our outlier detection work.
3. An analysis of the coverage of outliers from different methods to examine the consistency of these methods. Our results show that the outliers by the kNN algorithm cover most of the outliers by other methods, making it the most effective.
4. An analysis of the effect of outliers on model building. The result of the analysis shows that outliers can degrade the predictive power of predictive models by increasing prediction errors.

The remainder of this paper is organised as follows. We present in Section 2 the collection and preprocessing of the sensed flotation data used in this study. In Section 3, we present the outlier detection methods used in this study. In Section 4, we present the results and findings of this work. Finally, we draw our conclusions in Section 5.

2. Dataset and Preprocessing

2.1. Collection of Sensed Flotation Data

The dataset used in this study was obtained from a copper rougher flotation plant in south Australia. Figure 1 illustrates a schematic flow chart of the flotation operation. A primary rougher flotation stage receives feed input from a conditioning cell, scavenger concentrate, and cleaner tailings. The rougher concentrate undergoes further flotation in a cleaner stage to enhance concentrate grade, while the rougher tailings are directed to a scavenger stage. The final concentrate and tailings are derived from the cleaner concentrate and scavenger tailings, respectively [43].

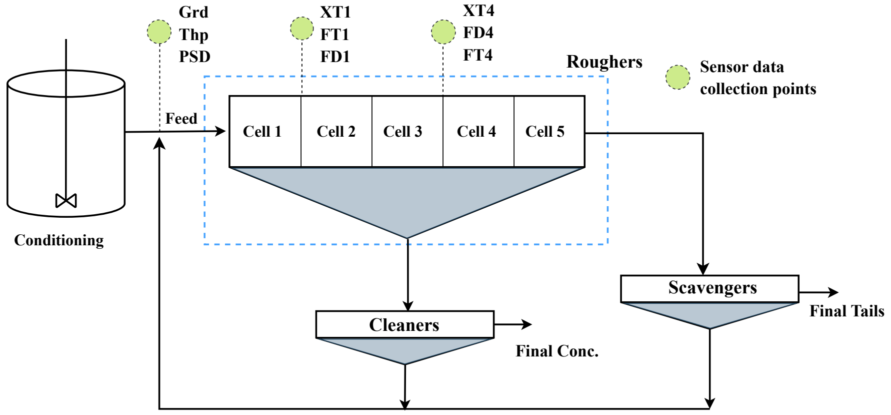


Figure 1. Schematic flow chart of the copper flotation operation indicating sensed data collected observations on rougher flotation stage. Grd—feed grade, Thp—throughput, PSD—% particle size passing 75 µm, XT1—xanthate dosage in cell 1, FT1—frother dosage in cell 1, FD1—froth depth in cell 1, XT4—xanthate dosage in cell 4, FT4—frother dosage in cell 4, FD4 - froth depth in cell 4.

The rougher flotation is a pivotal stage in the operation that reaches approximately 50%–60% recovery. An effective analysis of outlier detection of data from this stage can significantly help to detect operational errors early, improve process control, reduce chemical consumption, and reduce the loss of valuable minerals to tailings [44,45]. Given that the output of the rougher flotation stage influences key decision making and overall process performance, we analysed data from this stage in this study.

Table 1 shows the copper rougher flotation dataset consisting of ten input variables recorded every minute and a corresponding outcome variable. The outcome variable (i.e., rougher recovery) which indicates the performance of the operation was obtained from an Online Stream Analyser (OSA) results and computed using the expression in Equation (1). The dataset consists of 20,000 observations. Figure 2 visualises variations in the copper rougher flotation input variables. The x-axis shows the time index of each observation, and the y-axis represents the *normalised* data values from the original values scaled to a range between $[0, 1]$ using Equation (2).

$$Recovery, Rec = \left(\frac{c}{f}\right)\left(\frac{f-t}{c-t}\right)\times 100\% \tag{1}$$

where
c = rougher concentrate grade;
f = rougher feed grade;
t = rougher tailings grade.

The dataset consists of 20,000 observations. Figure 2 visualises variations in the copper rougher flotation input variables. The x-axis shows the time index of each observation and the y-axis represents the *normalised* data values from the original values scaled to a range between $[0, 1]$ using Equation (2).

$$x_i^{\text{norm}} = \frac{x_i - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{2}$$

where x_i^{norm} is the normalised value of x_i, x_i is the original value, x_{min} is the minimum value in the dataset, and x_{max} is the maximum value in the dataset.

Table 1. Nomenclature of flotation variables and their respective notations.

Variables		Location	Units	Notations
Input	Feed grade		%	Grd
	Throughput		t/h	Thp
	% Particle size passing 75 µm		%	PSD
	Xanthate dosage	Cell 1	mL/min	XT1
	Xanthate dosage	Cell 4	mL/min	XT4
	Frother dosage	Cell 1	mL/min	FD1
	Frother dosage	Cell 4	mL/min	FD4
	Froth depth	Cell 1	mm	FD1
	Froth depth	Cell 2/3 *	mm	FD2
	Froth depth	Cell 4/5 △	mm	FD4
Outcome	Recovery		%	Rec

* cell levels of cell 2 and 4 represent cell 3 and 5, respectively. △ Froth depth of cell 2 and 4 represents cell 3 and 5, respectively.

2.2. Preprocessing of Sensed Flotation Data

Data from industrial operations often have quality issues stemming from improper instrument calibrations or processes operating under quasi-stable conditions [12]. A meticulous cleaning process is essential to acquire accurate operational data values for analysis. Before conducting the outlier detection experiment, we cleaned the sensed flotation data by removing records with missing and wrong values. For example, in a typical flotation operation, a zero record for variables such as feed grade, throughput, and particle size distribution implies that there is no feed material (ore) in the plant. This scenario is highly implausible, since operating an empty plant serves no purpose. Therefore, records with zero data values that are indicative of operational instabilities such as shutdown and maintenance periods were excluded from the dataset.

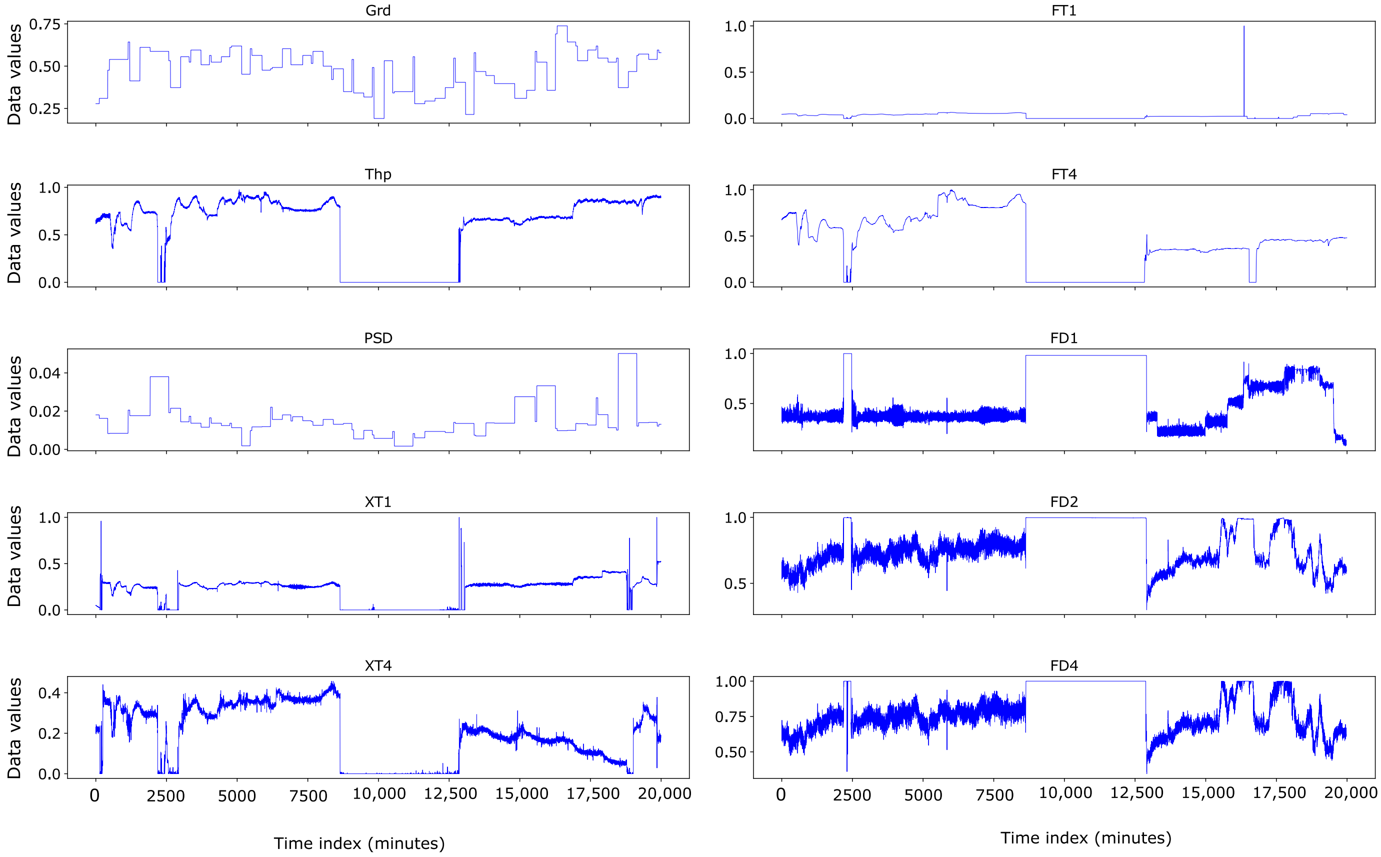


Figure 2. Visualisation of variations in the copper rougher flotation input variables. Grd—feed grade, Thp—throughput, PSD—% particle size passing 75 µm, XT1—xanthate dosage in cell 1, FT1—frother dosage in cell 1, FD1—froth depth in cell 1, FD2—froth depth in cell 2, XT4—xanthate dosage in cell 4, FT4—frother dosage in cell 4, FD4—froth depth in cell 4.

We illustrate this phenomenon using one of the input variables from the sensed flotation dataset: throughput. Throughput is a measure of the quantity of ore processed within a given time. When the throughput is zero, it indicates several possibilities: either there is no feed in the plant or the plant is not operational. However, the effect extends beyond the absence of feed, such that when the throughput is turned back on, there is a period during which the plant is settling into operation. For example, when the throughput drops to zero and remains off for 60 min, all observations within this period for all input variables should be excluded from the dataset during experimental analysis. Further, when the feed is turned back on, there is a period (usually three residence times, about 90 min) where circulating loads, reagent additions, air and level controls are slowly returning to equilibrium [2]. Observations within this period should be excluded from the experimental analysis, as they do not accurately reflect the flotation behaviour. Figure 3 shows the visualisation of the throughput variable in the copper rougher flotation data analysed in this study. Instances of zero throughput data values, indicated by orange-coloured dots, can be observed continuously over a long period of time in the dataset. Simply removing these observations would disrupt the continuity of the time sequence. Therefore, we tag these observations and exclude them from further analysis in this study.

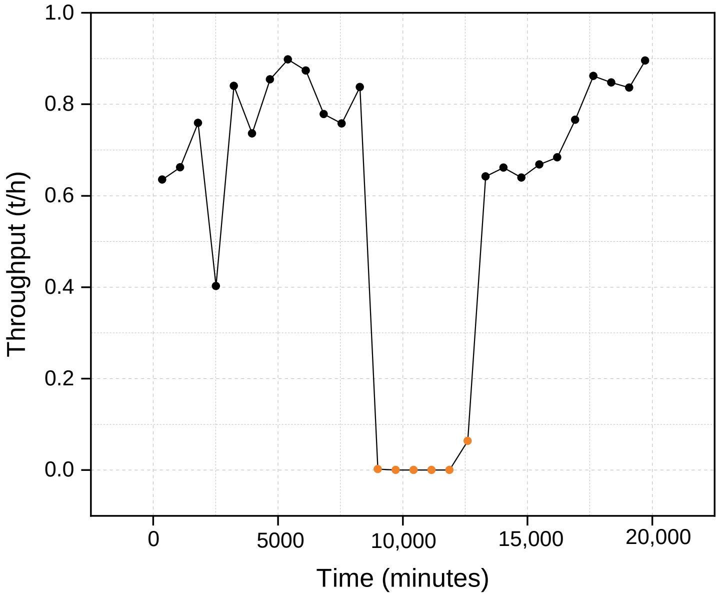


Figure 3. Visualisation of throughput of the copper rougher flotation data highlighting observations with zero data values.

Table 2 shows the descriptive statistics of the dataset excluding the tagged observations.

Table 2. Basic descriptive statistics.

Variable	Mean	Std	Min	Max
Grd	0.501713	0.103416	0.214286	0.738095
Thp	0.762795	0.10295	0.400435	0.974973
PSD	0.017179	0.009804	0.001816	0.050147
XT1	0.270034	0.083872	0.000000	1.000000
XT4	0.246805	0.111945	0.000000	0.457447
FT1	0.036745	0.01972	0.000000	1.000000
FT4	0.566525	0.207371	0.000000	1.000000
FD1	0.425439	0.186854	0.075309	0.998814
FD2	0.721567	0.123625	0.298425	0.996372
FD4	0.73869	0.115472	0.345048	1.000000
Rec	0.828509	0.062056	0.695473	0.944154

Std = standard deviation, Min = minimum, Max = maximum. Grd—feed grade, Thp—throughput, PSD—particle size distribution, XT1—xanthate dosage in cell 1, FT1—frother dosage in cell 1, FD1—froth depth in cell 1, XT4—xanthate dosage in cell 4, FT4—frother dosage in cell 4, FD4—froth depth in cell 4.

3. Methodology

In this section, we present the outlier detection methods used in this study in Sections 3.1–3.3. We formerly describe our method of identifying quasi-outliers and validating them in Sections 3.4 and 3.5. The selected machine learning (ML) methods include k-Nearest Neighbour (kNN), Local Outlier Factor (LOF), and Isolation Forest (ISF). The kNN method leverages the distances between an observation and its neighbours to detect outliers. LOF assesses outliers by comparing the local densities of observations with that of their neighbours. ISF identifies outliers by analysing the number of steps required to isolate observations from others using an ensemble of decision trees.

3.1. kNN

*k*NN [46] is a widely used technique for outlier detection in data mining. It determines outliers based on the distance of an observation to its nearest neighbours (*k*-distance). Distances are computed using metrics such as Euclidean, Manhattan, and Mahalanobis. A score is computed for each observation as the ratio of the sum of the distances to its nearest neighbours and the k value (Equation (3)). Observations with scores beyond a specified (*user-defined*) threshold are flagged as outliers.

$$S_i = \frac{1}{k}\Sigma_{j=1}^{k}\text{dist}\left(x_i, x_{(j)}\right) \tag{3}$$

where
S_i = score;
x_i = observation;
$x_{(j)}$ = the nearest neighbour jth of x_i;
k = number of nearest neighbours of x_i.

3.2. LOF

The Local Outlier Factor (LOF) method [34] is a density-based approach for detecting outliers. It compares the local density of an observation to the local densities of its neighbours. The primary idea is that outliers will have significantly lower local densities compared to their neighbours. Consider a dataset in n-dimensional space. For any point p in this space, we define k-dist(O) as the distance from O to its k-th nearest neighbour and $N_k(p)$ as the set of k-nearest neighbours of O. For points p and O, the Euclidean distance expressed in Equation (4) is typically used:

$$d(p,O) = \sqrt{\sum_{i=1}^{n}(p_i - O_i)^2} \tag{4}$$

The reachability distance between points p and O is defined in Equation (5) as

$$RD_k(p, O) = max\{k\text{-}dist(O),\ \ d(p,O)\} \tag{5}$$

This ensures that points within the same neighbourhood have similar reachability distances. For a point p, its Local Reachability Density (LRD) is defined as the inverse of the average reachability distance to its k-nearest neighbours given by Equation (6):

$$\text{LRD}_k(p) = \left(\frac{1}{|N_k(p)|}\sum O \in N_k(p)\text{RD}_k(p,O)\right)^{-1} \tag{6}$$

The Local Outlier Factor (LOF) of a point p is calculated as the average ratio of the LRD of p's neighbours to the LRD of p itself expressed in Equation (7):

$$\text{LOF}_k(p) = \frac{1}{|N_k(p)|}\sum O \in N_k(p)\frac{\text{LRD}_k(O)}{\text{LRD}_k(p)} \tag{7}$$

In interpreting LOF values, if $\text{LOF}_k(p) \approx 1$, p has a similar density to its neighbours. If $\text{LOF}_k(p) > 1$, p has a lower density than its neighbours, indicating it might be an outlier. The higher the LOF value, the more likely p is an outlier. In Figure 4, point O represents the observation being evaluated. Points $p1$, $p2$, $p3$, and $p4$ are potential k-nearest neighbours of O. The dashed circle represents the k-distance(O), which is the distance to the k-th nearest neighbour. $d(p,O)$ shows the distance between O and one of its neighbours (in this case, $p2$). The LOF method provides a robust way to identify outliers by considering the local context of each data point. It is particularly useful in datasets with varying densities, where global density-based methods might fail to detect local outliers. The method's ability to provide a degree of outlierness, rather than a binary classification, allows for more careful analysis and decision making in outlier detection tasks.

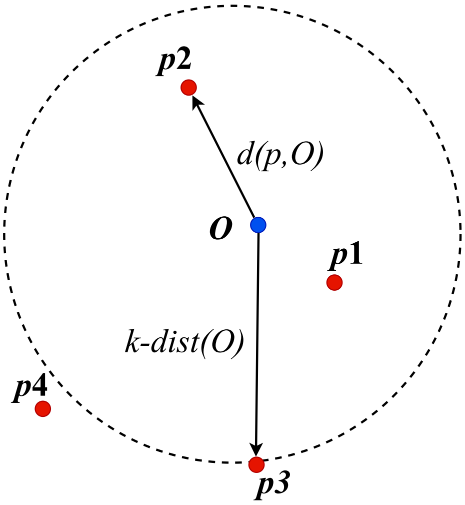


Figure 4. Illustration of LOF concepts.

3.3. ISF

The Isolation Forest (IF) [47] is a method for detecting anomalies based on the principle of isolating observations. This approach is founded on the idea that anomalies are both rare and distinct, making them more susceptible to isolation than normal points in a dataset. Consider a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the i-th observation, and $\mathbf{X}$ is the matrix containing all n observations. The IF algorithm constructs a forest of isolation trees (iTrees), which are each built using a subsample of size n from X. The construction of an iTree proceeds as follows: at each node, an attribute q is randomly selected from the d available features. A split value p is then randomly chosen between $\max(\mathbf{x}q)$ and $\min(\mathbf{x}q)$ for the selected attribute. This process creates an internal node with the test condition $\mathbf{x}_q < p$. The algorithm recursively applies this procedure to build left and right subtrees using the resulting subsets. The recursion terminates when one of three conditions is met: (i) the tree reaches a specified height limit, (ii) the node contains only one instance, or (iii) all instances at the node have identical attribute values. The anomaly score for a point $\mathbf{x}$ is derived from its average path length $E[h(\mathbf{x})]$ across the forest of t iTrees. The score is formally defined as

$$s(\mathbf{x}, n) = 2^{-\frac{E[h(\mathbf{x})]}{c(n)}} \tag{8}$$

where $c(n)$ is a normalisation factor given by:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{9}$$

In this equation, $H(i)$ represents the harmonic number, which can be approximated by $\ln(i) + 0.578$ (Euler's constant) [48]. The score $s(\mathbf{x}, n)$ exhibits several important properties. As $s(\mathbf{x}, n)$ approaches 1, the likelihood of $\mathbf{x}$ being an anomaly increases. Conversely, when $s(\mathbf{x}, n)$ is significantly less than 0.5, $\mathbf{x}$ is more likely to be a normal instance. Scores around 0.5 indicate instances that are neither clearly anomalous nor clearly normal. To employ IF for anomaly detection, a threshold $s_0 \in [0, 1]$ is established. Instances for which $s(\mathbf{x}, n) > s_0$ are classified as anomalies. The selection of s_0 is crucial and depends on the specific requirements of the application, balancing the need to detect true anomalies against the risk of false positives.

3.4. The 2σ Rule

True outliers are important for the evaluation of the effectiveness of the above algorithms and the impact of outliers on predictive models. However, true outliers are generally not available in real applications unless the experts in the applications label them based on application contexts. For this reason, in research, synthetic true outliers are often implanted into the dataset for detection [49].

True outliers are unknown in the dataset from our study. Generated synthetic outliers may not be suitable for the application context. We utilise a statistical distribution principle to establish a metric *tr* (to be elaborated upon shortly) for outlier identification. Specifically, within the distribution of all observations with respect to the metric, any observation lying beyond 2 standard deviations from the mean of the metric *tr* is designated as an outlier. The outliers identified by this method are referred to as *quasi-outliers* in this study. Based on a normal distribution guideline, observations that are 2σ away from the mean count as 5% of total observations. We note that if true outliers are known for example from experts, they should be used to replace quasi-outliers. Quasi-outliers are not the same as true outliers, and this will be demonstrated later in the experimental section. Nevertheless, we use quasi-outliers as a baseline to compare ML outlier detection algorithms.

3.5. Trend Differential tr(i,j) and Quasi-Outliers

Determining thresholds for identifying outliers is vital. In this section, we introduce our proposed approach based on the trend differential, denoted as $tr(i,j)$, which is computed for every element in the dataset. The trend differential is used to identify observations that significantly deviate from the expected trend, potentially marking them as outliers.

3.5.1. Standard Deviation Factor

Given a score vector $\mathbf{s} = [s_1, s_2, \ldots, s_n]^T \in \mathbb{R}^n$, with mean μ_s and standard deviation σ_s, the standard deviation factor for the *i*-th element s_i is denoted as $sf(i, \mathbf{s})$. It is defined as

$$sf(i, \mathbf{s}) = \frac{|s_i - \mu_s|}{\sigma_s}, \quad i = 1, 2, \ldots, n \tag{10}$$

This factor $sf(i, \mathbf{s})$ quantifies the deviation of s_i from the mean μ_s in terms of the standard deviation σ_s. The factor indicates the rarity of the value s_i within the vector $\mathbf{s}$. Conventionally, if $sf(i, \mathbf{s}) > 2$, the observation s_i is considered to be in the 5th percentile and is flagged as a potential outlier. For a large sample size, we assume the estimated scores are close to a normal distribution and adopt 2σ for thresholding the score column. Scaling to 1.5σ or 1σ would capture many normal observations as outliers leading to false detection. Adopting higher thresholds, for example, 2.7σ or 3σ would capture less observations, missing outlier observations in the data.

3.5.2. Trend Differential Calculation

For a given dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{n \times d}$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$, the trend differential $t_r(i,j)$ for each element x_{ij} is computed to detect significant deviations in trends across the dataset. The trend differential $t_r(i,j)$ is defined as the maximum of two absolute differences:

$$tr(i,j) = \max\Big(|x_{ij} - x_{(i-1)j}|, |x_{ij} - x_{(i+1)j}|\Big) \tag{11}$$

This differential $tr(i,j)$ captures the most significant local change in the data trend at the specific point (i,j) relative to its neighbouring points.

3.5.3. Total Score Factor

After computing the trend differentials $tr(i,j)$ for all elements, the standard deviation factor sf is applied to each column of the matrix *tr* (comprising the trend differentials). The total score factor $ttlsf(i)$ for the *i*-th row is then computed by aggregating the standard deviation factors across all columns.

The total score factor is defined as:

$$ttlsf(i) = \sqrt{\frac{1}{d}\sum_{j=1}^{d}(sf(i, tr(*, j)))^2} \tag{12}$$

Here, $ttlsf(i)$ represents the root mean square of the standard deviation factors for the trend differentials across all dimensions j for the i-th observation. This provides a comprehensive measure of the overall deviation from the expected trend in the entire row.

3.5.4. Quasi-Outliers

Observations are identified as quasi-outliers if their total score factor exceeds a specified threshold. Typically, a standard deviation factor $sf(i) > 2$ is used as a threshold to identify quasi-outliers. These are data points that exhibit significant but not extreme deviations from the norm, which makes them candidates for further analysis as potential anomalies.

3.6. Cover Rate (CR)

To assess the effectiveness of the algorithms, we obtained a ranking of the quasi-outliers and determined their coverage by the detection algorithms. By coverage, we mean a quasi-outlier is considered covered by an algorithm g if the record index i of the quasi-outlier is found among the outliers O of g or a sequential neighbour i' of i is in O. The concept of sequential neighbours refers to observations that are close to each other in a sequential or ordered dataset. In this context, they must be within [i−Δ, i+Δ] where Δ is a time range to reflect the fact that the flotation response to an event may take up to 20–30 min. We investigated various Δ values from 3, 5, 10, and 15 min to determine the best coverage by the detection algorithm. We calculate the cover rate of the detection algorithms using Equation (13) as

$$CR = \frac{N_{cu}}{N_{qo}} \tag{13}$$

where
N_{cu} is the number of quasi-outliers covered.
N_{qo} is the total number of quasi-outliers.

3.7. Assessing the Impact of Outliers on Prediction Performance

The predictive model utilised in this study is Extreme Gradient Boosting (XGBoost), which is a highly efficient and scalable implementation of gradient boosted decision trees. We developed XGBoost models to predict the output variable (Rec) using the ten (10) input variables previous described in Table 1. The XGBoost models were tuned using the optimal hyperparameters described in Table 3.

Table 3. XGBoost model hyperparameter settings.

Description	Parameter
Base learner	Gradient boosted trees
Learning objective	Regression with squared error
Regularisation lambda	2.0
Maximum depth of trees	2

The performance metrics used for evaluation were the root mean square error (RMSE), the mean absolute percentage error (MAPE), and the coefficient of determination (R^2) defined as

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{14}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{16}$$

where n is the total number of observations, y_i is the actual output for the i-th observation, and $\hat{y}_i$ is the predicted output for the i-th observation, and $\bar{y}$ is the mean of the actual output variables. Higher R^2 values indicate better model performance, while lower RMSE and MAPE values indicate better performance.

3.7.1. Quasi-Outlier Removal Analysis

We first investigated model performance by systematically removing different levels of quasi-outliers from the dataset. The dataset was split into training (80%) and testing (20%) subsets. Models were trained with varying degrees of quasi-outlier removal and then evaluated on test datasets.

3.7.2. Outlier-Inclusive vs. Outlier-Exclusive Model Comparison

In the second approach, we employed the following methodology: we applied the kNN outlier score to rank the dataset, employing a 2σ threshold to differentiate between outliers and normal observations. We then randomly selected and reserved 1000 samples as an independent test set. The remaining data were partitioned into training (80%) and validation (20%) subsets with the latter serving to assess the model's generalisation capacity and mitigate potential underfitting or overfitting. We developed two distinct models: Model 1, trained on the complete dataset including outliers, and Model 2, trained exclusively on data with outliers removed. Both models were subsequently evaluated using the reserved test set with performance quantified through root mean square error (RMSE), mean absolute percentage error (MAPE), and coefficient of determination (R^2) metrics. This systematic approach, illustrated in Figure 5, enabled a rigorous comparison of model performance with and without the influence of outlier observations.

All experiments were conducted on a personal computer with Intel(R) Core (TM) i5-10210U CPU @ 4 GHz and 8 GB memory with a Windows 10 operating system. The outlier detection algorithms were sourced from Scikit-learn open source libraries, except for the trend differential, and implemented using Python programming software, version 3.12.0 for data preprocessing, experiments, and result analysis.

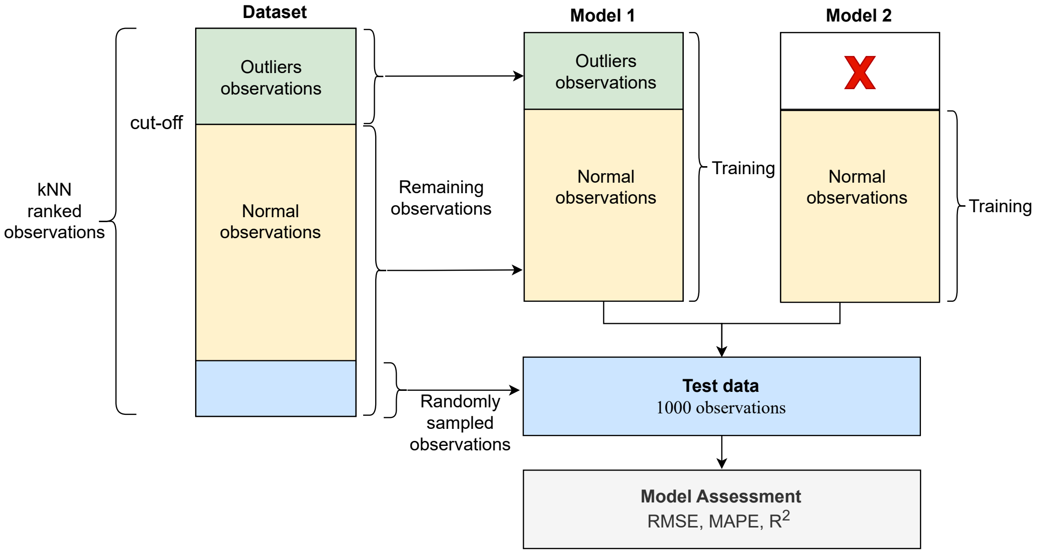


Figure 5. Flow diagram illustrating model performance calculation with and without outliers.

4. Results

In this section, we present the outcomes of our experiments for the outlier detection algorithms presented in the previous section. Our objective is to analyse the effectiveness of these algorithms in detecting true outliers in the flotation data. In the following, we first present the results of neighbourhood size for kNN in Section 4.1. Next, we present in Section 4.2 the quasi-outliers detected by the trend differential method and their properties through visualisation. Then, we compare the effectiveness of the outlier detection algorithms in Section 4.3. Sections 4.4–4.6 highlights covered and uncovered quasi-outliers as well as non-covering outliers from the detection algorithms. Finally, we present the results demonstrating the impact of outliers on prediction performance in Section 4.7

4.1. Nearest Neighbourhood Size

The neighbourhood size k is a user-defined parameter representing the number of nearest neighbours to be considered in calculating the outlier score. It is an important parameter in the identification of outliers. If $k = 1$, all observations receive the same outlier score of 0, and if k equals the total number of records of the dataset, all observations are from the same distance. The appropriate k value can help differentiate rare observations from other observations [50,51]. By leveraging Euclidean distances, we use the elbow graph method [52] to optimise the parameter k using weighted averaging.

For each k from 3 to 100, the average error that predicts the outcome is calculated for all observations following the prediction of k nearest neighbour. From k and its average error, a graph is plotted. The elbow point is the inflexion point at which the down-trend of the line is changing to the horizontal trend.

Equations (17) and (18) are based on the following definitions:

- $\text{dist}_i[j,0]$: Euclidean distance for the j-th nearest neighbour of the i-th observation.
- $\text{dist}_i[j,1]$: Corresponding target value $y[j]$ for the j-th nearest neighbour.

We define

$$\mathbf{Y}_i = \sum_{j=0}^{k} \text{dist}_i[j,0] \cdot \text{dist}_i[j,1] \tag{17}$$

$$\mathbf{S}_i = \sum_{j=0}^{k} \text{dist}_i[j,0] \tag{18}$$

where $\mathbf{Y}_i$ and $\mathbf{S}_i$ represent the weighted sum and sum of distances for the i-th observation.

The error for each observation i is computed as the absolute difference between the actual target value $y[i]$ and the weighted average:

$$\text{error}_i = \left| y[i] - \frac{\mathbf{Y}_i}{\mathbf{S}_i \cdot (k+1)} \right| \tag{19}$$

This error is then accumulated to obtain a total error for the current k:

$$\text{TotalError} = \sum_i \text{error}_i \tag{20}$$

By deriving the total errors for each k, we can identify the optimal k that minimises the total error. The results are visualised by plotting the corresponding total errors against k, facilitating a clear identification of the optimal k. This approach provides a robust framework for determining the optimal k for the flotation data used in this study.

Figure 6 shows an elbow graph of error measures against k-distance values. The curve takes a bend for k values greater than 20 and plateaus as the neighbourhood size increases. According to the graph, we adapt a neighbourhood size of $k = 20$ in this study for the detection of outliers in the sensed flotation dataset.

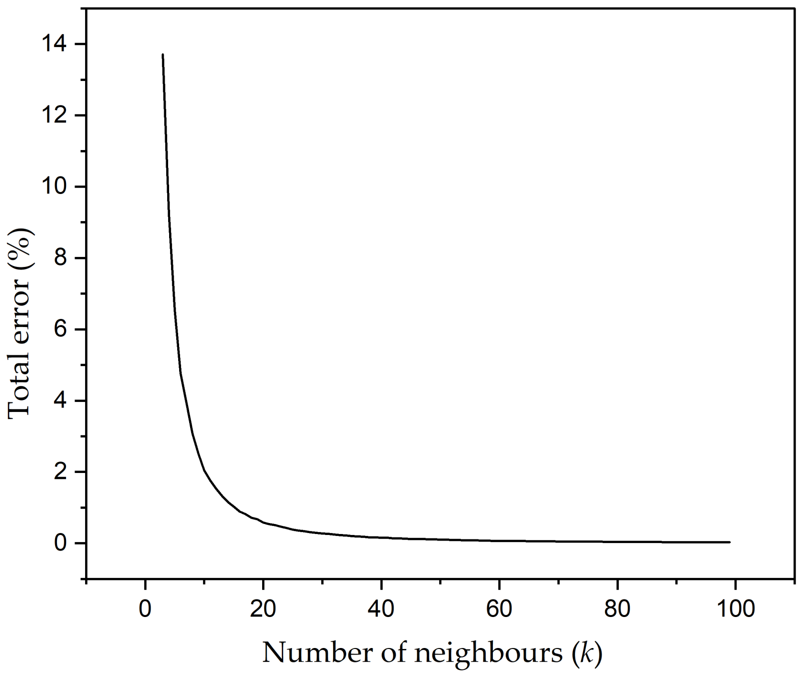


Figure 6. Selection of nearest neighbourhood size (k) for the copper rougher flotation dataset.

4.2. *Quasi-Outliers*

Based on the definition of quasi-outliers in Equation (12), 150 quasi-outliers were obtained from the dataset. Figure 7 shows a plot of the first 50 quasi-outliers on every feature variable where the y-axis is the normalised feature value and the x-axis is the time when the observation was taken. The feature values are plotted in blue and the quasi-outliers are plotted in red-coloured vertical lines, which indicate the time dimension when the outlier occurs. It can be seen that the outliers exist mostly in regions where the data peaks or drops and reflect across some input variables.

For easier reading, we use Figure 8 to show a few quasi-outlier observations among their 20 nearest neighbours. The intuition is that a normal observation would follow the clusters of their nearest neighbours, whereas an outlier observation would deviate from the cluster of it's nearest neighbours. In Figure 8, the red line is the quasi-outlier observation, and the blue lines are the 20 nearest neighbours to the quasi-outlier observation. The x-axis displays the input variables labelled Grd, Thp, PSD, XT1, XT4, FT1, FT4, FD1, FD2, and FD4, and the y-axis shows the normalised data values of the observations across the input variables.

It can be seen that the quasi-outlier observations deviate from the clusters of their neighbours, which are captured within some of the input variables. Significant deviations can be seen in the input variables XT1, XT4, FT1, FT4, Grd, and FD4. In our application, the variable(s) where the deviation occurs can be inspected by operators or experts to determine what may be causing the production of the erroneous observations.

4.3. *Effectiveness of the Outlier Detection Algorithms*

We now assess the effectiveness of the outlier detection algorithms to detect quasi-outliers in the dataset. Although the individual outlier algorithms detect quasi-outliers, they do not rank them equally. This means that top-ranked quasi-outliers may not be ranked among the top outliers by other algorithms. This makes sense, as each algorithm follows a different principle in its detection. However, it can be expected that the algorithms would rank quasi-outliers in the topmost outliers and non-outliers at the bottom [53]. We refer to observations that are ranked as the topmost outliers by the detection algorithms as the '*worst outliers*'. We analyse the effectiveness of the outlier detection algorithms using the cover rate presented in the following section.

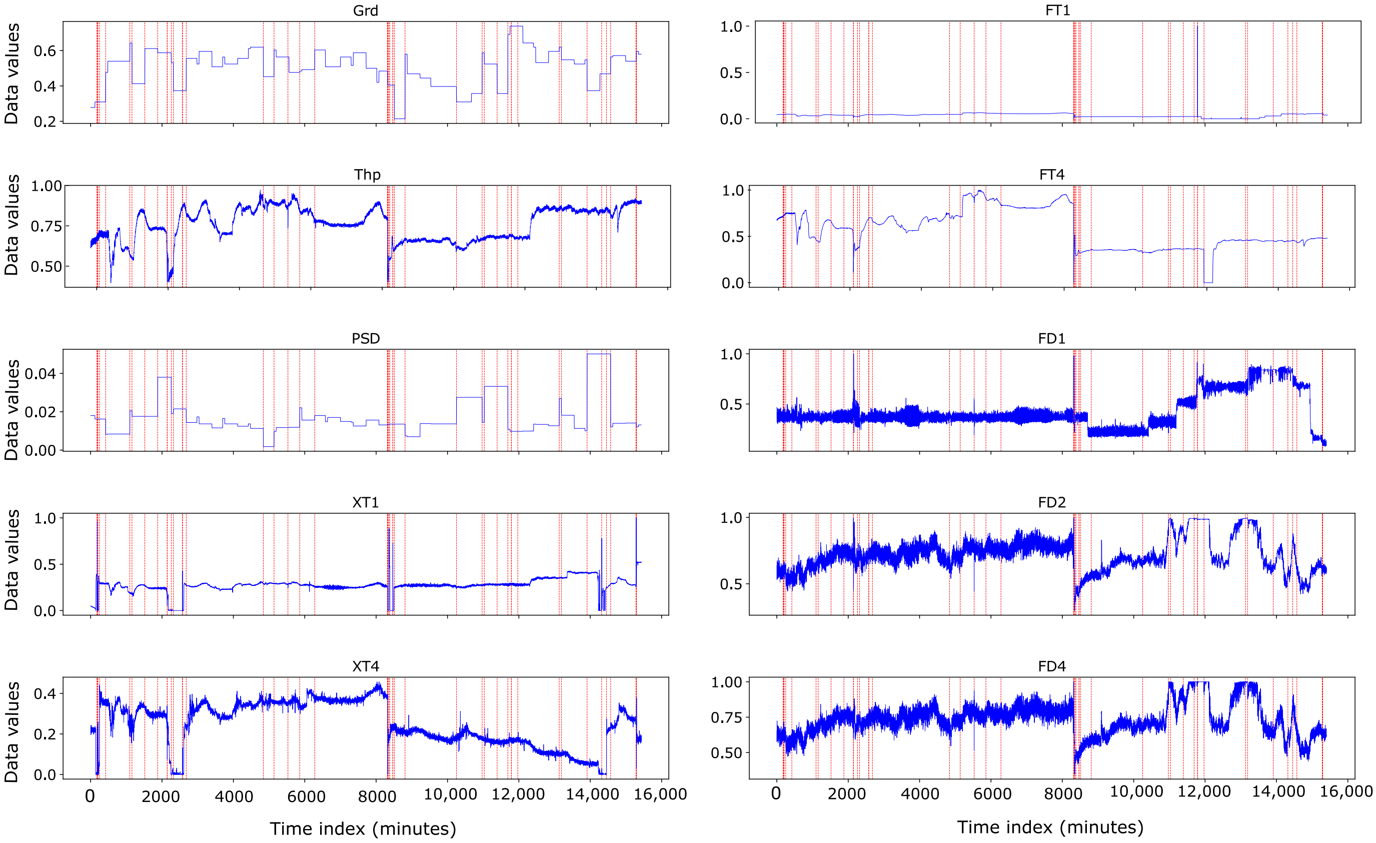


Figure 7. Visualisation of first 50 quasi-outliers on each input variable along the time dimension. Note: Grd—feed grade, Thp—throughput, PSD—% particle size passing 75 µm, XT1—xanthate dosage in cell 1, FT1—frother dosage in cell 1, FD1—froth depth in cell 1, FD2—froth depth in cell 2, XT4—xanthate dosage in cell 4, FT4—frother dosage in cell 4, FD4—froth depth in cell 4.

Cover Rate (CR)

We use Figure 9 to show the coverage of the first 50 quasi-outliers by the detection algorithms. Quasi-outliers only are plotted in Figure 9a, and algorithm coverage is plotted for kNN in Figure 9b, LOF in Figure 9c, and ISF in Figure 9d. The plots show the observation indexes on the y-axis and their ranking on the x-axis. The red circles ('o') represent the quasi-outliers and the blue markers ('x') represent the outliers from the detection algorithms. It can be seen that several outliers from the detection algorithm completely cover the quasi-outliers around index 8000. In addition, few sequential neighbours can be observed around this index. Quasi-outliers around index 2000 and below show sequential neighbour coverage with the majority of them observed in kNN and ISF coverage. LOF coverage had the least sequential neighbours around this index. It is worth mentioning that around index 2000, most of the sequential neighbours from kNN coverage achieved the best ranking of the three algorithms. Similarly, quasi-outliers above index 10,000 realised only three sequential neighbours coverage from each algorithm. The three sequential neighbours of kNN coverage above 10,000 index rank the quasi-outliers as top outliers (with smaller ranking value), which is followed by LOF and then ISF. This means that the kNN covers most of the quasi-outliers best, which indicates a better detection compared to LOF and ISF.

We present in Table 4 the ranking of the first 50 quasi-outliers. The first column represents the quasi-outlier observations with their corresponding ranking in the second column. The next three columns show their rank coverage positions for kNN, LOF, and ISF. Quasi-outliers that are not covered within $\Delta[-10,+10]$ of a detection algorithm are assigned an $*$ for the rank position.

In Table 5, we present the *CR* of the detection algorithms for different cover ranges. From the results, we observe the following:

1. kNN obtained the highest *CR*, which was followed by LOF and then ISF across all the ranges investigated with maximum *CR* values of 0.84, 0.65, 0.64 for kNN, LOF, and ISF, respectively,
2. When the time range ∆ reaches 10, the coverage stabilises and does not improve any further.

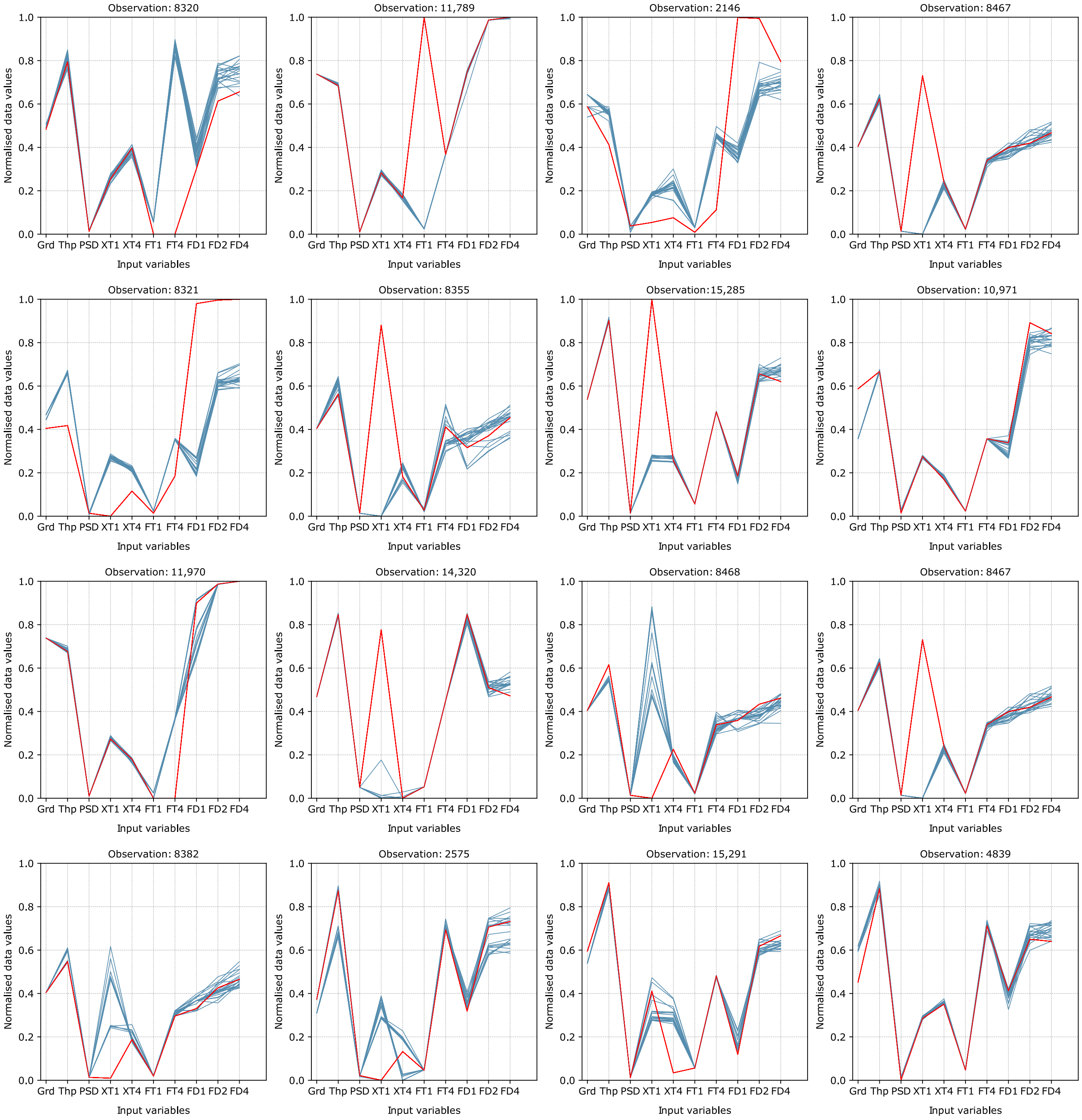
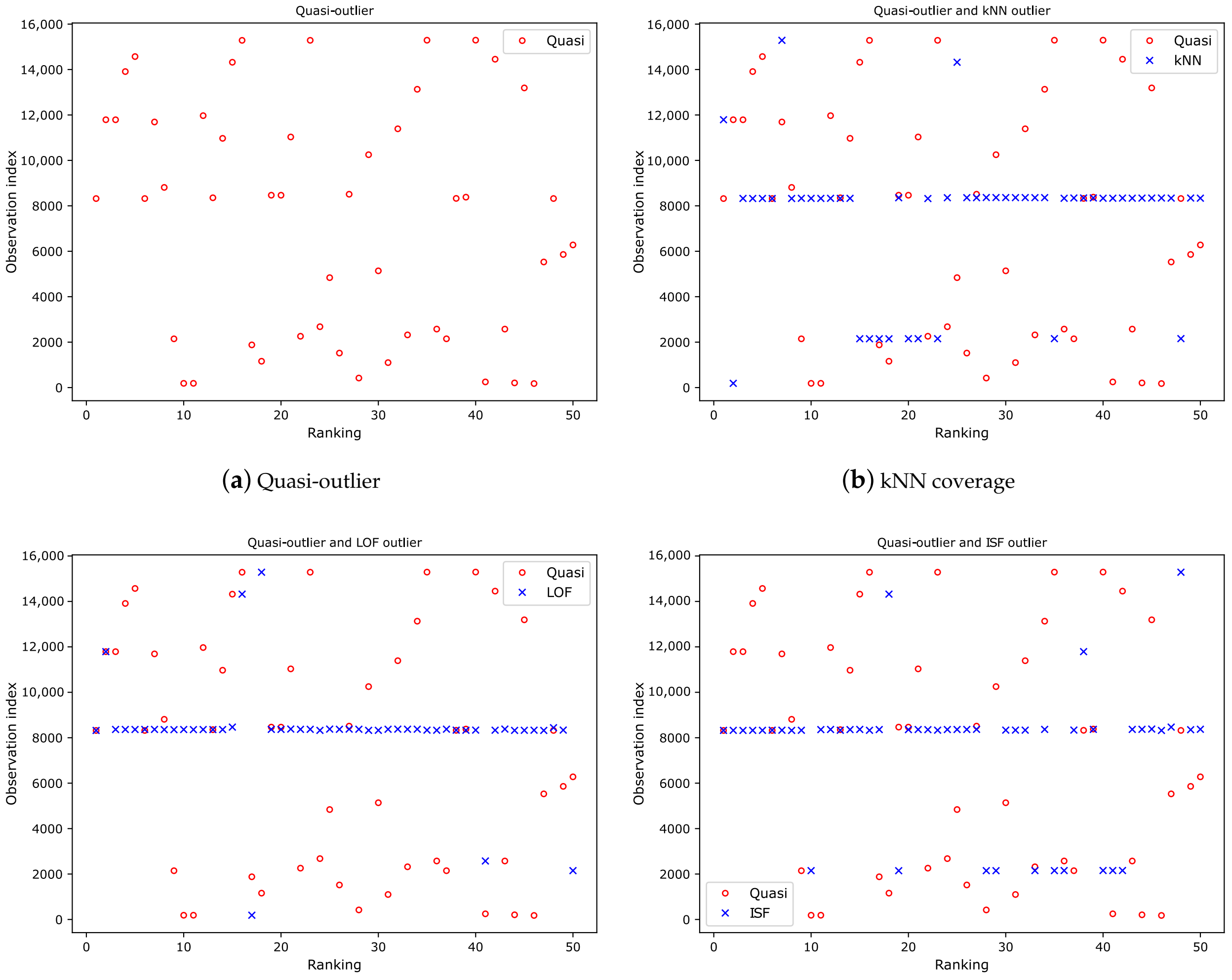


Figure 8. Feature plots of quasi-outliers identified by the various outlier detection algorithms. Note: The red line represents an observation under consideration, blue lines represent the 20 nearest neighbours of the observation. Grd—feed grade, Thp—throughput, PSD—% particle size passing 75 µm, XT1—xanthate dosage in cell 1, FT1—frother dosage in cell 1, FD1—froth depth in cell 1, FD2—froth depth in cell 2, XT4—xanthate dosage in cell 4, FT4—frother dosage in cell 4, FD4—froth depth in cell 4.



(**a**) Quasi-outlier (**b**) kNN coverage

(**c**) LOF coverage (**d**) ISF coverage

Figure 9. Coverage plot. The red circles ('o') represent the quasi-outliers and the blue markers ('x') represent the outliers from the detection algorithms.

Table 4. Quasi-outlier ranking within $\Delta[-10, +10]$ by the detection algorithms.

Quasi-Outlier	Ranking	Ranking Positions		
		kNN Position	LOF Position	ISF Position
8320	1	3	1	1
11,790	2	1	2	38
11,789	3	1	2	38
13,911	4	208	*	398
14,571	5	315	1220	*
8321	6	3	1	1
11,691	7	*	*	*
8811	8	*	223	*
2146	9	15	50	10
186	10	2	17	54
187	11	2	17	54
11,970	12	102	546	102
8355	13	19	3	11
10,971	14	431	148	938
14,320	15	25	16	18
15,285	16	7	18	48

Table 4. *Cont.*

Quasi-Outlier	Ranking	Ranking Positions		
		kNN Position	LOF Position	ISF Position
1879	17	*	*	*
1159	18	271	218	*
8468	19	57	15	47
8467	20	57	15	47
11,031	21	943	*	*
2259	22	235	880	266
15,286	23	7	18	48
2679	24	369	314	*
4839	25	455	473	662
1519	26	*	*	*
8511	27	92	54	81
420	28	233	*	*
10,251	29	957	266	*
5139	30	*	*	*
1099	31	438	*	*
11,391	32	*	*	*
2319	33	141	*	494
13,131	34	*	*	*
15,291	35	2	18	48
2575	36	7	41	75
2147	37	15	50	10
8327	38	3	1	1
8382	39	59	21	45
15,293	40	7	18	48
249	41	164	365	459
14,454	42	278	534	346
2574	43	74	41	75
206	44	132	320	199
13,191	45	*	*	142
177	46	2	17	54
5529	47	80	70	77
8322	48	2	1	1
5859	49	1041	*	*
6279	50	*	*	*

Note: A lower number means an algorithm considers an observation the worst outlier, whereas a higher number means an algorithm considers the observation as less suspicious of being an outlier. * denotes quasi-outliers not covered by the corresponding algorithm.

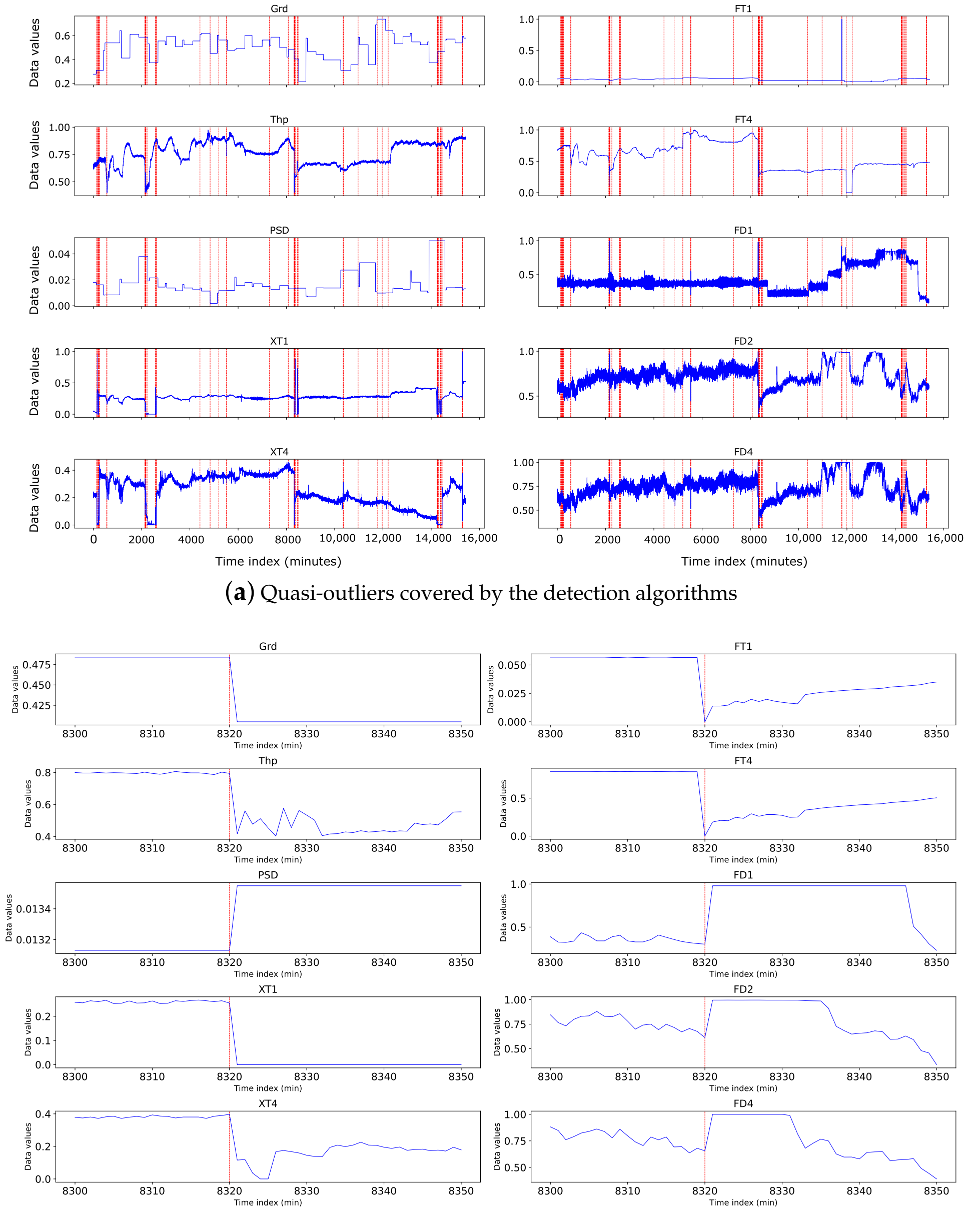
Table 5. Cover rate of quasi-outliers by detection methods.

Algorithms	Cover Range and Rate of Outlier Detection Algorithms			
	$\boldsymbol{\Delta[-3,3]}$	$\boldsymbol{\Delta[-5,5]}$	$\boldsymbol{\Delta[-10,10]}$	$\boldsymbol{\Delta[-15,15]}$
kNN	**0.77**	**0.79**	**0.84**	**0.84**
LOF	0.59	0.61	0.65	0.65
ISF	0.57	0.61	0.63	0.64

Note: Bold text represent highest values which indicate better cover rate (CR) and effective detection.

4.4. Covered Quasi-Outliers

We show in Figure 10 quasi-outliers that are covered by the detection algorithms. We note that these outliers represent true or confirmed outliers and should not be ignored. Figure 10a shows all outliers covered by the detection algorithms (plotted in red). The outliers mark the time dimensions where significant deviations occur, which are characterised by peaks and jumps across all the input variables. We present in Figure 10b the worst covered outlier observation showing jumps across the input features. This means that they cause rippling effects across multiple variables, leading to an extensive compromise of the observation and the outcome it generates.



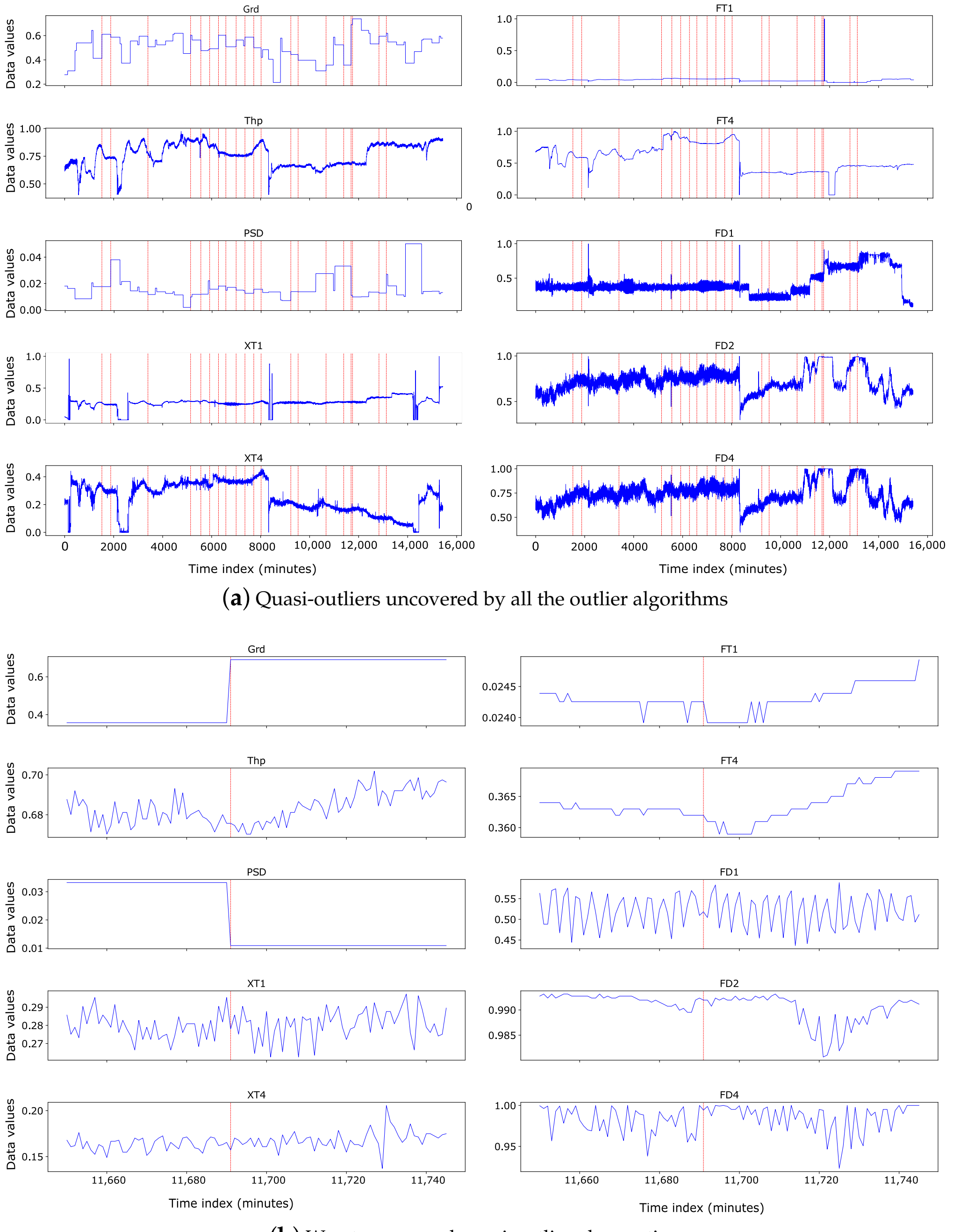
(**a**) Quasi-outliers covered by the detection algorithms

(**b**) Worst covered quasi-outlier observation

Figure 10. Time series plot of quasi-outliers (**a**) covered by the detection algorithms and (**b**) worst covered quasi-outlier observation. The red lines represent the outlier observations at the specific time index, and the blue lines represent normalised data values.

4.5. Uncovered Quasi-Outliers

Now, we analyse the quasi-outliers uncovered by the detection algorithms. We plot all uncovered quasi-outliers in Figure 11a. We found that all these uncovered quasi-outliers have the property of a one-sided trend jump. A one-sided trend means that before the time of the outlier observation, variables take similar values; at the time of the outlier, the values of some variables have either jumped up or down. After the outlier time index, the variables take similar values again (but maybe at new value levels for some of them). This phenomenon can be observed in Figure 11b with a one-sided jump before the observation in variable Grd and a one-sided drop before the observation in variable PSD. A normal outlier has a two-sided trend change to make the observation different from others, making the observation detected by other algorithms. The observation at the one-sided trend change is either similar to its neighbours before the observation or similar to its neighbours after the observation, and this similarity indicates that the outliers at the one-sided trend change are falsely identified by the trend differential algorithm. As a result, quasi-outliers uncovered by kNN can be ignored.



(**a**) Quasi-outliers uncovered by all the outlier algorithms

(**b**) Worst uncovered quasi-outlier observation

Figure 11. Time series plot of (**a**) all uncovered quasi-outliers, and (**b**) worst quasi-outlier not covered by the detection algorithms.The red lines represent the outlier observations at specific time index and the blue lines represent normalised data values.

4.6. Non-Covering kNN, LOF, and ISF Outliers

The following question is whether outliers detected by kNN, LOF, and ISF which do not cover any quasi-outlier, are important, or should they be ignored? We analyse the outliers from the detection algorithms and present in Table 6 the number of non-covering outliers. We represent non-covering outliers from kNN, LOF, and ISF by kNN_{nco}, LOF_{nco}, and ISF_{nco}, respectively. We found 80 outliers as kNN_{nco}, 11 as LOF_{nco}, and 2 as ISF_{nco}. The analysis also revealed that all $ISF_{nco} \in kNN_{nco}$ and the majority of the $LOF_{nco} \in kNN_{nco}$. As such, we considered only kNN_{nco} and LOF_{nco} in further analysis.

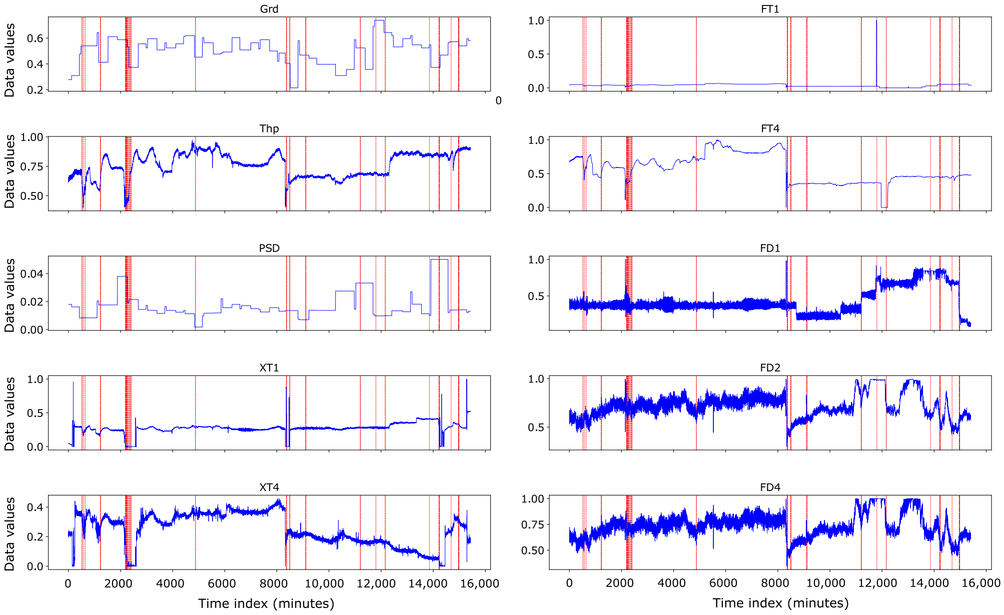
Table 6. Non-covering outliers from the detection algorithms.

Algorithms	Number of Outliers (2σ Threshold)	Number of Non-Covering Outliers
kNN	300	80
LOF	74	11
ISF	28	2

Figure 12 visualises kNN_{nco} in the time dimension for all variables. Figure 12a shows all the kNN_{nco}, whereas Figure 12b shows the worst kNN_{nco} observations.

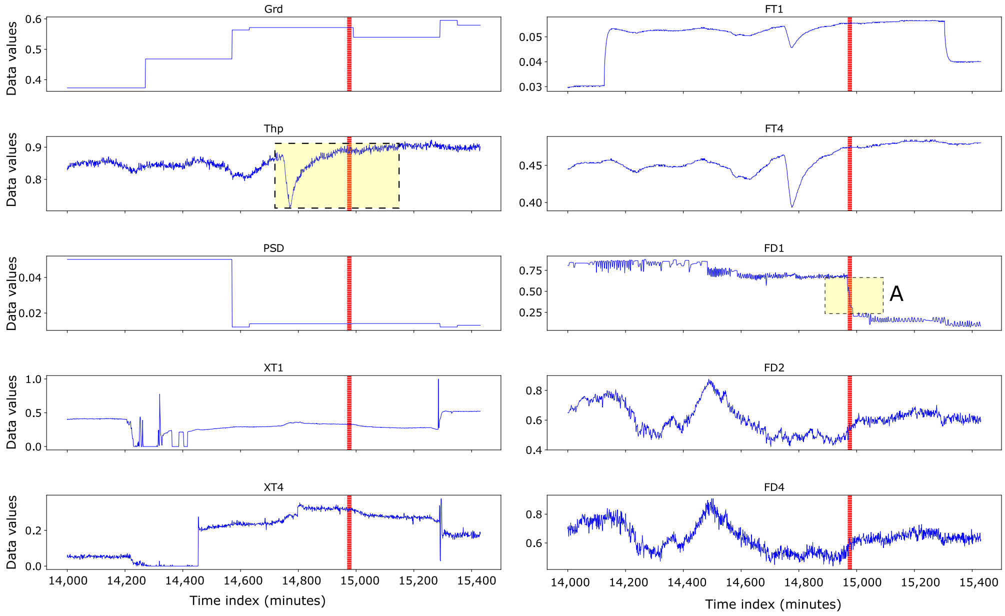
We use Figure 12b to show how these outliers are validated. Considering the group A of outliers near time = 15,000 in Figure 12b, their sequential neighbours are coloured yellow in the Thp and FD1 subplots. Then, 2.5 h before the outlier (near time = 14,800), the subplots reflecting inputs (Grd, PSD, XT1, XT4, FD2, and FD4) are stable except for Thp and FT4, which have deep dives and less dive for FT1. These dives can be associated with disturbances in the flotation system such as the throughput being turned off, instability of the froth due to high depth, resulting in the froth collapse; or just a change in the air feeding. As is well known, the impact of a change in inputs to the flotation fades away within 0.5–1.5 h, and the system should be stable. However after about 2 h (near time = 15,000), most of the input variables come back to stable values, and we expect that the variable FD1 should remain stable as well. However, FD1 shows a sudden drop as shown near time = 15,000. The sudden drop in FD1 for this group of observations represents an unexpected change. All the non-covering kNN outliers (in Figure 12a) demonstrate a similar 'unexpected change' property. We capture these suspicious observations as outliers.

Similarly, we visualise LOF_{nco} in Figure 13. Figure 13a shows all LOF_{nco} values, whereas Figure 13b shows the worst LOF_{nco} observations. In Figure 13b, the yellow rectangle shows the time dimension and variable (XT1) where the worst LOF_{nco} occurs. The observations in this region have a density relatively lower than that of their neighbours. As such, the LOF algorithm rightly detects them as outliers.



(**a**) All kNN_{nco} observations

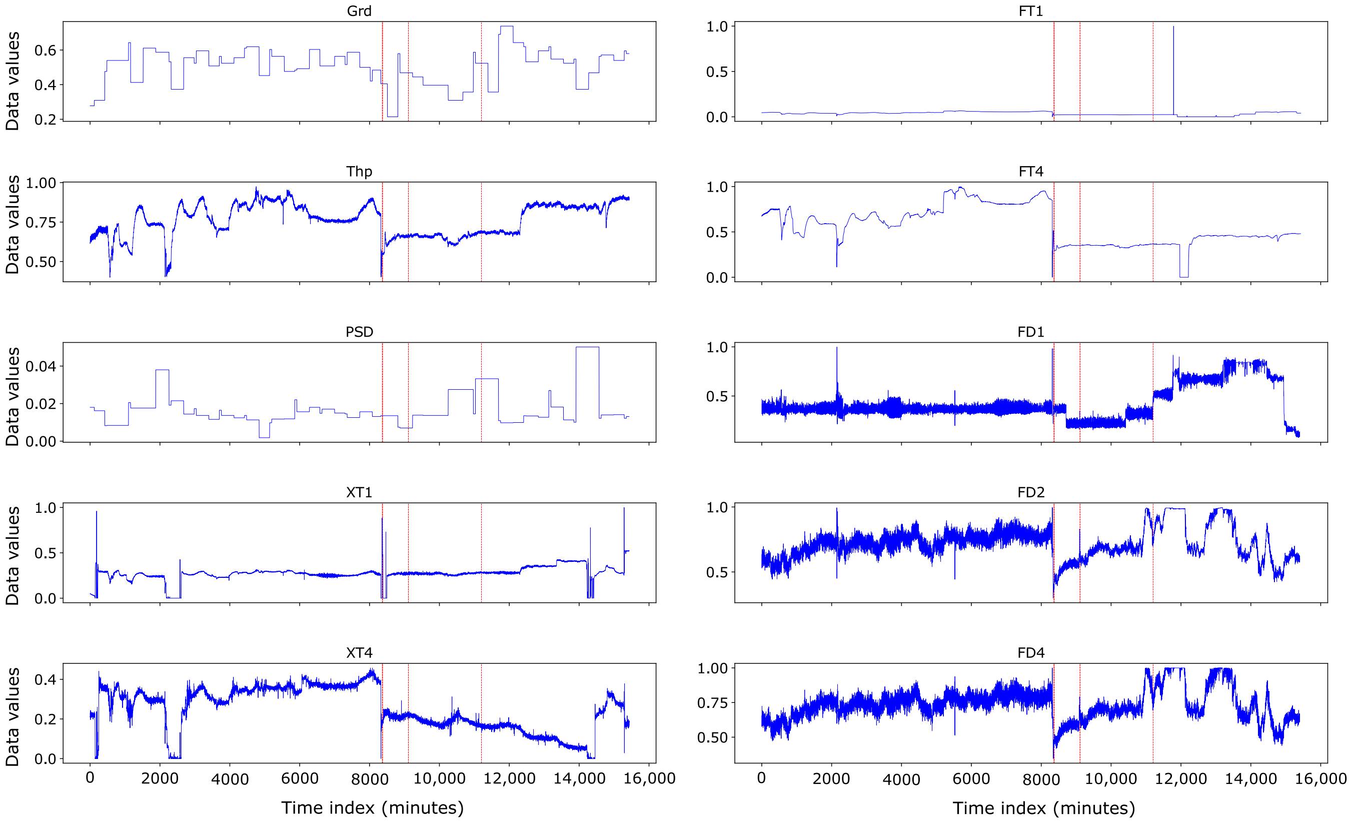
Figure 12. *Cont.*



(**b**) Worst kNN_{nco} observations

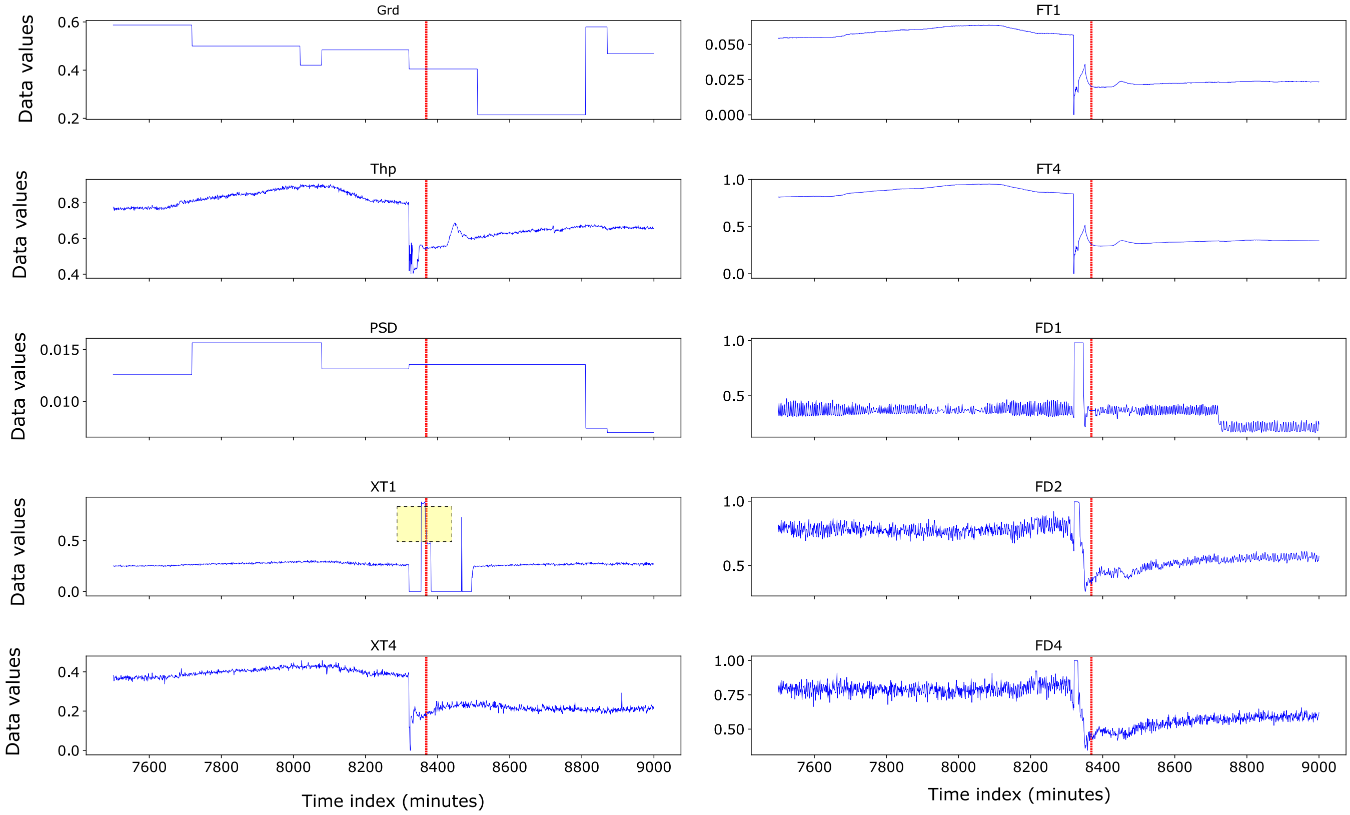
Figure 12. Time series plot of non-covering kNN outliers. The red lines represent the outlier observations at a specific time index and the blue lines represent normalised data values.

We conclude that both kNN_{nco} and LOF_{nco} observations should not be ignored. In our application, we recommend that such outliers should be carefully inspected in consultation with the domain knowledge of operation.



(**a**) All LOF_{nco} observations

Figure 13. *Cont.*



(b) Worst LOF_{nco} observations

Figure 13. Time series plot of LOF_{nco} observations. The red lines represent the outlier observations at the specific time index, and the blue lines represent normalised data values.

We summarise the experimental results as follows:

1. The common outliers by kNN and trend differential method are obvious outliers, and the trend differential method helps validate these.
2. The kNN non-covering outliers exhibit variations that correspond to system instabilities that could be linked to several complex interactions in the flotation system.
3. kNN identifies almost all outliers, while LOF adds a few subtle ones. In our application, we recommend that such outliers should not be ignored; rather, they should be carefully inspected.

4.7. Impact of Outliers on Prediction Performance

So far, we have argued that the method of deriving quasi-outliers is effective in identifying suspicious observations, and the outlier algorithms have confirmed worst quasi-outliers in the dataset. We present in Table 7 the impact of quasi-outliers on prediction performance. The results show different levels of quasi-outlier removal on the model performance assessment for both training and testing datasets. Overall, increasing quasi-outlier removal leads to a reduction in prediction errors and more accurate predictions. In this case, a mean absolute percentage error (MAPE; Equation (15)) of 0.62 % was achieved for predictions on test data containing quasi-outliers. After removing the top quasi-outliers, the prediction error for the test data decreased to about 0.23 %, which is approximately one third of the error when the quasi-outliers were included. This indicates that outliers can impact predictive model performance and must be carefully treated.

Table 7. Summary of the impact of removing quasi-outliers on prediction.

Quasi-Outliers Removed	Training		Testing	
	RMSE	MAPE (%)	RMSE	MAPE (%)
0 *	0.0083	0.6146	0.0079	0.6153
93	0.0028	0.2124	0.0036	0.2393
150	0.0026	0.1976	0.0031	0.2252

* Dataset in which no quasi-outlier observations have been removed.

The performances of Model 1 and Model 2 are presented in Tables 8 and 9, respectively. The performance of a model trained without outliers (Model 2) was better than the model with outliers (Model 1) for training, validation and testing. Model 1 achieved an RMSE (Equation (14)) of 0.0050, MAPE of 0.4715 %, and R^2 (Equation (16)) of 0.98, whereas Model 2 had an RMSE of 0.0040, MAPE of 0.4072 %, and R^2 of 0.99 when tested on the 'unseen' test data. This indicates that the outliers cause higher prediction errors and negatively impact the prediction performance.

Table 8. Model (Model 1) performance assessment with outliers.

Metrics	Training	Validation	Testing
RMSE	0.1125	0.1128	0.0050
MAPE (%)	0.8462	0.8711	0.4715
R^2	0.96	0.96	0.98

Table 9. Model (Model 2) performance assessment without outliers.

Metrics	Training	Validation	Testing
RMSE	0.0105	0.0111	0.0040
MAPE (%)	0.8193	0.8231	0.4072
R^2	0.97	0.97	0.99

5. Conclusions

This study introduced a novel 'trend differential' approach combined with a 2σ standard deviation factor to identify quasi-outliers in industrial flotation data. The effectiveness of this method was then validated using established outlier detection algorithms (kNN, LOF, and ISF). While our approach successfully captured a majority of the most significant outliers in the dataset, it is important to critically examine the implications and limitations of these findings. The visualisation of quasi-outliers revealed significant trend breaks across multiple variables, suggesting that our method can detect complex, multivariate anomalies. This aligns with previous research by Hodge and Austin [54], who emphasised the importance of considering multiple dimensions in outlier detection for industrial processes. However, the precise nature of these trend breaks and their root causes in the flotation process warrant further investigation.

Our introduction of a 5 % control limit to capture rare observations proved effective in identifying outliers, but it is crucial to consider the potential trade-offs. As pointed out by Aggarwal [55], there is always a risk of misclassifying legitimate rare events as outliers, which could lead to a loss of valuable information in process optimisation. Future work should explore adaptive thresholding techniques that can adjust to varying process conditions, as suggested by Liu et al. [56]. The observation that outliers occur in diverse directions within the dataset underscores the complexity of flotation processes and the challenges in outlier detection. This multidimensional nature of outliers aligns with findings by Markou and Singh [57], who highlighted the need for sophisticated, context-aware outlier detection methods in complex industrial settings.

Our evaluation of model prediction performance with and without outliers demonstrated their significant impact on prediction accuracy. While this supports the importance of outlier detection and removal for accurate modelling, it also raises questions about the potential loss of important process information. As cautioned by Rousseeuw and Hubert [58], an indiscriminate removal of outliers can lead to model overfitting and reduced generalisability.

The limitation of this study to three outlier detection algorithms, while providing valuable insights, also highlights the need for a more comprehensive comparison of methods. Future work could explore the application of deep learning techniques, such as autoencoders [59], which have shown promise in handling high-dimensional data typical in industrial processes. Moreover, the potential for real-time outlier detection and its integration into process control systems remains an exciting avenue for future research. As suggested by Ge et al. [60], the development of adaptive, online outlier detection methods could significantly enhance process monitoring and control in mineral processing operations. While our '*Trend differential*' approach shows promise in identifying complex outliers in flotation data, its practical implementation requires a careful consideration of process-specific factors and potential information loss. Future research should focus on developing more adaptive, context-aware outlier detection methods and exploring their integration with robust modelling techniques to enhance both the accuracy and interpretability of flotation process models.

The following conclusions can be drawn from this study:

- The outlier detection algorithms are effective in enhancing data quality, and their performance was assessed. The kNN algorithm performed best compared to LOF and ISF in terms of the number of quasi-outliers detected and covered, as kNN ranks the majority of the worst outliers as top outliers. The effectiveness of the detection algorithms can be ordered as $kNN > LOF > ISF$.
- Training data containing outliers can cause predictive models to make larger errors on non-outlier input records. The study showed that outliers have detrimental effects on prediction performance compared to 'normal' observations. This negative impact of outliers should not be overlooked as they produce inaccurate performance outcomes, especially in high-dimensional data.
- The dynamic nature of flotation processes makes distinguishing 'normal' observations from outliers complex. Analysts should avoid rigidly applying predetermined thresholds for outlier detection without thorough investigations and consultation with industry experts. It is essential to assess the degree of outlier behaviour in flotation data using both analytical methods and domain knowledge to enhance data quality.

This research is highly significant to both the research community and the mineral processing industry. It demonstrates that unsupervised ML algorithms are effective in analysing data from flotation operations. These algorithms can detect outliers, enhance data quality for predictive analysis, and improve process optimisation for future planning and decision making.

Author Contributions: Conceptualisation, C.L. and J.L.; methodology, C.L.; software, C.L.; validation, C.L., J.L. and C.G.; formal analysis, C.L. and J.L.; investigation, C.L. and J.L.; resources, C.L., J.L. and R.K.A.; data curation, C.L., J.L. and R.K.A.; writing—original draft preparation, C.L.; writing—review and editing, C.L., J.L., R.K.A., C.G. and M.Z.; visualisation, C.L. and J.L.; supervision, J.L., R.K.A., C.G. and M.Z.; project administration, W.S.; funding acquisition, W.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Australian Research Council Integrated Operations for Complex Resources Industrial Transformation Training Centre (project number IC190100017) and funded by universities, industry and the Australian Government.

Data Availability Statement: Data are not available for confidentiality reasons.

Conflicts of Interest: The authors declare no conflicts of interest. Christopher Greet is an employee of Magotteaux Australia Pty. Ltd. The paper reflects the views of the scientist and not the company.

Abbreviations

The following abbreviations are used in this manuscript:

ML	machine learning
kNN	k-Nearest Neighbour
LOF	Local Outlier Factor
ISF	Isolation Forest
XGBoost	Extreme Gradient Boosting
LRD	local reachability distance
RMSE	root mean square error
MAPE	mean absolute percentage error
tr	trend differential
CR	Cover rate
kNN_{nco}	kNN non-covering outliers
LOF_{nco}	LOF non-covering outliers
ISF_{nco}	ISF non-covering outliers

References

1. Pawlik, M. Fundamentals of froth flotation. *ChemTexts* **2022**, *8*, 19. [CrossRef]
2. Wills, B.A.; Finch, J.A. Froth flotation. In *Wills' Mineral Processing Technology: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery*, 8th ed.; Elsevier: Amsterdam, The Netherlands, 2015; Chapter 12, pp. 265–380. [CrossRef]
3. Dixon, W.J. Analysis of extreme values. *Ann. Math. Stat.* **1950**, *21*, 488–506. Available online: https://www.jstor.org/stable/2236602 (accessed on 25 July 2024). [CrossRef]
4. Devavarapu, Y.; Bedadhala, R.R.; Shaik, S.S.; Pendela, C.R.K.; Ashesh, K. Credit Card Fraud Detection Using Outlier Analysis and Detection. In Proceedings of the 2024 4th International Conference on Intelligent Technologies (CONIT), Bali, Indonesia, 21–23 June 2024; pp. 1–7. [CrossRef]
5. Zhang, J.; Zulkernine, M. Anomaly based network intrusion detection with unsupervised outlier detection. In Proceedings of the 2006 IEEE International Conference on Communications, Istanbul, Turkey, 11–15 June 2006; IEEE: Piscataway, NJ, USA, 2006; Volume 5, pp. 2388–2393. [CrossRef]
6. Mall, S.; Srivastava, A.; Mazumdar, B.D.; Mishra, M.; Bangare, S.L.; Deepak, A. Implementation of machine learning techniques for disease diagnosis. *Mater. Today Proc.* **2022**, *51*, 2198–2201. [CrossRef]
7. Jemwa, G.T.; Aldrich, C. Kernel-based fault diagnosis on mineral processing plants. *Miner. Eng.* **2006**, *19*, 1149–1162. [CrossRef]
8. Hawkins, D.M. *Identification of Outliers*; Monographs on Applied Probability and Statistics; Chapman and Hall: London, UK, 1980; pp. 1–124. [CrossRef]
9. Smiti, A. A critical overview of outlier detection methods. *Comput. Sci. Rev.* **2020**, *38*, 100–306. [CrossRef]
10. Han, J.; Kamber, M. *Data Mining: Concepts and Techniques*, 2nd ed.; The Morgan Kaufmann Series in Data Management Systems; Morgan Kaufmann: San Francisco, CA, USA, 2006; p. 800. Available online: https://api.semanticscholar.org/CorpusID:195837802 (accessed on 25 July 2024).
11. Pahuja, D.; Yadav, R. Outlier Detection for Different Applications: Review. *Int. J. Eng. Res. Technol. (IJERT)* **2013**, *2*, 1–7. Available online: https://www.ijert.org/outlier-detection-for-different-applications-review (accessed on 25 July 2024).
12. Xu, S.; Lu, B.; Baldea, M.; Edgar, T.F.; Wojsznis, W.; Blevins, T.; Nixon, M. Data cleaning in the process industries. *Rev. Chem. Eng.* **2015**, *31*, 453–490. [CrossRef]
13. Estay, H.; Lois-Morales, P.; Montes-Atenas, G.; Ruiz del Solar, J. On the challenges of applying machine learning in mineral processing and extractive metallurgy. *Minerals* **2023**, *13*, 788. [CrossRef]
14. Hodouin, D.; Jämsä-Jounela, S.L.; Carvalho, M.; Bergh, L. State of the art and challenges in mineral processing control. *Control Eng. Pract.* **2001**, *9*, 995–1005. [CrossRef]
15. Greet, C.J.; Selga, K. Continuous, real-time pulp chemistry measurements and what they tell us about metallurgical performance. In Proceedings of the 48th Annual Canadian Mineral Processors Operators Conference, Ottawa, ON, Canada, 19–21 January 2016; pp. 154–163. Available online: https://www.onemine.org/documents/continuous-real-time-pulp-chemistry-measurements-and-what-they-tell-us-about-metallurgical-performance (accessed on 25 July 2024).
16. Greet, C.; Small, G.; Steinier, P.; Grano, S. The Magotteaux Mill®: Investigating the effect of grinding media on pulp chemistry and flotation performance. *Miner. Eng.* **2004**, *17*, 891–896. [CrossRef]
17. Li, C.; Gao, Z. Effect of grinding media on the surface property and flotation behavior of scheelite particles. *Powder Technol.* **2017**, *322*, 386–392. [CrossRef]
18. Hodouin, D. Methods for automatic control, observation, and optimization in mineral processing plants. *J. Process Control* **2011**, *21*, 211–225. [CrossRef]
19. Beckman, R.; Cook, R. Outlier.......... s. *Technometrics* **1983**, *25*, 119–149. [CrossRef]
20. Grubbs, F.E. Sample Criteria for Testing Outlying Observations. *Ann. Math. Stat.* **1950**, *21*, 27–58. [CrossRef]

21. Doerffel, K. Beurteilung von Analysenverfahren und-ergebnissen. *Fresenius J. Anal. Chem.* **1961**, *185*, 1–98. Originally published as "Fresenius' Zeitschrift für analytische Chemie". [CrossRef]
22. Peirce, B. Criterion for the Rejection of Doubtful Observations. *Astron. J.* **1852**, *2*, 161–163. Available online: https://adsabs.harvard.edu/full/1852AJ......2..161P (accessed on 25 July 2024). [CrossRef]
23. Lin, L.; Sherman, P.D. Cleaning Data the Chauvenet Way. In Proceedings of the SouthEast SAS Users Group (SESUG), Hilton Head, SC, USA, 4–6 November 2007; pp. 1–11. Paper SA11. Available online: https://analytics.ncsu.edu/sesug/2007/SA11.pdf (accessed on 25 August 2024).
24. Dastjerdy, B.; Saeidi, A.; Heidarzadeh, S. Review of Applicable Outlier Detection Methods to Treat Geomechanical Data. *Geotechnics* **2023**, *3*, 375–396. [CrossRef]
25. Davies, L.; Gather, U. The identification of multiple outliers. *J. Am. Stat. Assoc.* **1993**, *88*, 782–792. [CrossRef]
26. Hampel, F. The influence curve and its role in robust estimation. *J. Am. Stat. Assoc.* **1974**, *69*, 383–393. [CrossRef]
27. Siegel, A.F.; Morgan, C.J. *Statistics and Data Analysis: An Introduction*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 1996.
28. Tukey, J. *Exploratory Data Analysis*; Addison-Wesley: Reading, MA, USA, 1977; Volume 2, pp. 192–194. [CrossRef]
29. Knorr, E.M.; Ng, R.T. Algorithms for mining distance-based outliers in large datasets. In Proceedings of the 24th International Conference on Very Large Data Bases, New York, NY, USA, 24–27 August 1998; pp. 392–403. [CrossRef]
30. Kriegel, H.P.; Kröger, P.; Zimek, A. Outlier Detection Techniques, 2010. In Proceedings of the Tutorial at SIAM International Conference on Data Mining (SDM 2010), Columbus, OH, USA, 29 April–1 May 2010. Available online: https://imada.sdu.dk/u/zimek/publications/SDM2010/sdm10-outlier-tutorial.pdf (accessed on 25 August 2024).
31. Ramaswamy, S.; Rastogi, R.; Shim, K. Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 427–438. [CrossRef]
32. Tang, B.; He, H. A local density-based approach for outlier detection. *Neurocomputing* **2017**, *241*, 171–180. [CrossRef]
33. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. OPTICS-OF: Identifying local outliers. In Proceedings of the Principles of Data Mining and Knowledge Discovery: Third European Conference, PKDD'99, Prague, Czech Republic, 15–18 September 1999. [CrossRef]
34. Breunig, M.; Kriegel, H.; Ng, R.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dalas, TX, USA, 16–18 May 2000; pp. 93–104. [CrossRef]
35. Kriegel, H.P.; Kroger, P.; Schubert, E.; Zimek, A. Interpreting and unifying outlier scores. In Proceedings of the 2011 SIAM International Conference on Data Mining (SDM), Mesa, AZ, USA, 28–30 April 2011; pp. 13–24. [CrossRef]
36. De Vries, T.; Chawla, S.; Houle, M.E. Finding local anomalies in very high dimensional space. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, NSW, Australia, 13–17 December 2010; pp. 128–137. [CrossRef]
37. Kriegel, H.P.; Kroger, P.; Schubert, E.; Zimek, A. LoOP: Local outlier probabilities. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November 2009; pp. 1649–1652. [CrossRef]
38. Papadimitriou, S.; Kitagawa, H.; Gibbons, P.B.; Faloutsos, C. LOCI: Fast outlier detection using the local correlation integral. In Proceedings of the 19th International Conference on Data Engineering, Bangalore, India, 5–8 March 2003; pp. 315–326. [CrossRef]
39. Agyemang, M.; Ezeife, C.I. LSC-Mine: Algorithm for mining local outliers. In Proceedings of the 15th Information Resource Management Association (IRMA) International Conference, Innovations Through Information Technology, New Orleans, LA, USA, 23–26 May 2004; pp. 5–8. Available online: https://www.irma-international.org/proceedingpaper/lsc-mine-algorithm-mining-local/32284/ (accessed on 25 July 2024).
40. Zhang, K.; Hutter, M.; Jin, H. A new local distance-based outlier detection approach for scattered real-world data. In Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2009), Bangkok, Thailand, 27–30 April 2009; pp. 813–822. [CrossRef]
41. Tang, J.; Chen, Z.; Fu, A.W.C.; Cheung, D.W. Enhancing effectiveness of outlier detections for low density patterns. In Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2002), Taipei, Taiwan, 6–8 May 2002; pp. 535–548. [CrossRef]
42. Zhang, J.; Yang, Y. Density-Distance Outlier Detection Algorithm Based on Natural Neighborhood. *Axioms* **2023**, *12*, 425. [CrossRef]
43. Amankwaa-Kyeremeh, B.; Zhang, J.; Zanin, M.; Skinner, W.; Asamoah, R.K. Feature selection and Gaussian process prediction of rougher copper recovery. *Miner. Eng.* **2021**, *170*, 107041. [CrossRef]
44. Ghodrati, S.; Nakhaei, F.; VandGhorbany, O.; Hekmati, M. Modeling and optimization of chemical reagents to improve copper flotation performance using response surface methodology. *Energy Sources Part Recover. Util. Environ. Eff.* **2020**, *42*, 1633–1648. [CrossRef]
45. Yianatos, J.; Carrasco, C.; Bergh, L.; Vinnett, L.; Torres, C. Modelling and simulation of rougher flotation circuits. *Int. J. Miner. Process.* **2012**, *112–113*, 63–70. [CrossRef]
46. Knorr, E.M.; Ng, R.T. A Unified Notion of Outliers: Properties and Computation. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, USA, 14–17 August 1997; pp. 219–222. https://cdn.aaai.org/KDD/1997/KDD97-044.pdf.
47. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422. [CrossRef]

48. Lesouple, J.; Baudoin, C.; Spigai, M.; Tourneret, J.Y. Generalized isolation forest for anomaly detection. *Pattern Recognit. Lett.* **2021**, *149*, 109–119. [CrossRef]
49. 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]
50. Boehmke, B.; Greenwell, B.M. K-means Clustering. In *Hands-On Machine Learning with R*, 1st ed.; Chapman and Hall/CRC: Boca Raton, FL, USA, 2019; Chapter 20, pp. 339–415. [CrossRef]
51. Xu, H.; Zhang, L.; Li, P.; Zhu, F. Outlier detection algorithm based on k-nearest neighbors-local outlier factor. *J. Algorithms Comput. Technol.* **2022**, *16*, 1–12. [CrossRef]
52. Yuan, C.; Yang, H. Research on K-value selection method of K-means clustering algorithm. *J* **2019**, *2*, 226–235. [CrossRef]
53. Huang, H.; Mehrotra, K.; Mohan, C.K. Rank-based outlier detection. *J. Stat. Comput. Simul.* **2013**, *83*, 518–531. [CrossRef]
54. Hodge, V.J.; Austin, J. A Survey of Outlier Detection Methodologies. *Artif. Intell. Rev.* **2004**, *22*, 85–126. :AIRE.0000045502.10941.a9 [CrossRef]
55. Aggarwal, C.C. *Outlier Analysis*, 2nd ed.; Springer International Publishing: Cham, Switzerland, 2017. [CrossRef]
56. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation-Based Anomaly Detection. *ACM Trans. Knowl. Discov. Data* **2012**, *6*, 1–39. [CrossRef]
57. Markou, M.; Singh, S. Novelty detection: A review—Part 1: Statistical approaches. *Signal Process.* **2003**, *83*, 2481–2497. [CrossRef]
58. Rousseeuw, P.J.; Hubert, M. Anomaly detection by robust statistics. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1236. [CrossRef]
59. Chalapathy, R.; Chawla, S. Deep Learning for Anomaly Detection: A Survey. *arXiv* **2019**, arXiv:1901.03407.
60. Ge, Z.; Song, Z.; Gao, F. Review of Recent Research on Data-Based Process Monitoring. *Ind. Eng. Chem. Res.* **2013**, *52*, 3543–3562. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.