

Article

# A Graphical Calibration Method for a Water Quality Model Considering Process Variability Versus Delay Time: Theory and a Case Study

Eyal Brill and Michael Bendersky \* 

Faculty of Industrial Engineering and Technology Management, Holon Institute of Technology, Holon 5810201, Israel

\* Correspondence: michael.bendersky@gmail.com

**Abstract:** Process Variability (PV) is a significant water quality time-series measurement. It is a critical element in detecting abnormality. Typically, the quality control system should raise an alert if the PV exceeds its normal value after a proper delay time (DT). The literature does not address the relation between the extended process variability and the time delay for a warning. The current paper shows a graphical method for calibrating a Water Quality Model based on these two parameters. The amount of variability is calculated based on the Euclidean distance between records in a dataset. Typically, each multivariable process has some relation between the variability and the time delay. In the case of a short period (a few minutes), the PV may be high. However, as the relevant DT is longer, it is expected to see the PV converge to some steady state. The current paper examines a method for estimating the relationship between the two measurements (PV and DT) as a detection tool for abnormality. Given the user's classification of the actual event for true and false events, the method shows how to build a graphical map that helps the user select the best thresholds for the model. The last section of the paper offers an implementation of the method using real-world data.

**Keywords:** water quality model; graphical tool; tuning



**Citation:** Brill, E.; Bendersky, M. A Graphical Calibration Method for a Water Quality Model Considering Process Variability Versus Delay Time: Theory and a Case Study. *Computation* **2023**, *11*, 200. <https://doi.org/10.3390/computation11100200>

Academic Editors: Yudong Zhang and Francesco Cauteruccio

Received: 9 July 2023

Revised: 25 September 2023

Accepted: 30 September 2023

Published: 7 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Water Quality Model (WQM) is a term used to describe a mathematical method that enables predicting, controlling, and ensuring drinking water quality in a water distribution system. Such a method includes input water quality measurements, such as the pH, turbidity, TOC, redox, free chlorine, and others. As output, the process generates a decision indicating whether the water is safe for drinking. An algorithm or some computational method calculates the transition between inputs and output. One may perform the calculation manually or fully automated.

The ISO standard 24522 [1] (Water and wastewater quality event detection) defines several options for this type of model. Generally, all processes, including water quality measurement, have some PV (process variability). Regarding the DT (delay time), the PV may be high in the case of a short period (a few minutes). However, as the relevant DT is longer, it is expected to see the PV converge to some steady state. The current paper examines the relationship between the two measurements (PV and DT) as a detection tool for abnormality.

The literature includes many examples of identification methods for detecting water quality abnormality. The current section reviews the most common forms. Dempster [2] lists real-time ways of using machine learning to identify water quality abnormality. Dempster introduced the concept of multivalued mapping, which associates each element of a space of possibilities with a set of possible outcomes. The above method allows for a more flexible representation of uncertainty than traditional probability theory, where each factor is associated with a single result. Perelman et al. [3] used an artificial neural network to

predict water quality using time series data and compared the expected value to the actual one to detect water quality disruptions.

The proposed process involves using a combination of multiple sensors and data fusion algorithms to obtain more accurate and reliable measurements of water quality parameters. The innovation is related to a new statistical method for identifying abnormal events in the water quality data, such as changes in the concentration of different contaminants, based on the correlation structure of the data. Dibo et al. [4] implemented an extended Dempster–Shafer method. They used the comprehensive Dempster–Shafer method, a mathematical framework for reasoning under uncertainty, to integrate the data from different sensors and determine the likelihood of contamination events. The authors combined data from multiple sensors, including temperature, pH, and dissolved oxygen, to comprehensively view the water quality in real-time. Liu et al. [5] improved the Dempster–Shafer method. Their main achievements are higher accuracy, cost-effective, and comprehensive monitoring.

Hagar et al. [6] used CANAY, a free software tool that uses a multivariable regression to predict water quality parameters and compare the expected value with the actual value measured. An event is declared when this difference is large enough for a significant period. [7] Demonstrated the usage of PCA (Principal Component Analysis) analysis applied to UV absorption as a tool for identifying possible water quality events, the authors used a UV-Vis absorption spectroscopy sensor to measure the absorbance spectrum of water samples at different wavelengths and then applied PPCA (Probabilistic Principal Component Analysis) to analyze the data and detect anomalies by calculating the likelihood of an example in the PPCA plan.

Since water distribution systems are chlorinated, the level of residual chlorine should be in some steady state. Deviation from this steady state signals abnormality. Nejjari et al. [8] showed a solution for this issue. They based their model on residual chlorine levels using the EPANET modeling tool.

In recent work, Mao et al. [9] implemented a spatial-temporal-based event detection approach with multivariate time-series data for water quality monitoring (M-STED). The third part of their method established a spatial model with Bayesian networks to estimate the state of the backbones in the next timestamp and trace any “outlier” node to its neighborhoods to detect a contamination event. The authors of [10] applied three models for comparing the actual versus the predicted value of the quality parameters. First was the Holt–Winters model, which uses a time series forecasting method that considers three components: level, trend, and seasonality. The model uses exponential smoothing to forecast past time series values. Second was the McKenna model, a mathematical model used for predicting the behavior of chemical reactions in liquid media. The model uses the mass balance equations that describe the conservation of the different species involved in the response. Third was an artificial neural network (ANN). This method uses the input of measurements to detect the gap between the actual and predicted values of the dependent output. They inspected events by observing unexpected changes in quality measurements compared to the predicted values. The authors of [11] extended the above approach by applying several additional models to water quality data. After preprocessing and feature selection, they used k-nearest neighbors (k-NN), support vector machines (SVM), decision trees, random forests, and ANNs to classify the water quality data as either standard or anomalous.

They used logistic regression, linear discriminant analysis, SVM, an ANN, deep neural network (DNN), recurrent neural network (RNN), and long short-term memory (LSTM). The results showed that the random forest algorithm outperformed the other models regarding the accuracy, sensitivity, and specificity in detecting anomalies in the water quality data.

Each of the above WQMs included tuning parameters. These parameters are the knobs used to make the model valuable and accurate. Accuracy refers to four situations: true positive, false positive, true negative, and false negative. The aggregation of these four

measurements is translated into a statistical quality index, such as Kappa, indicating the model's reliability.

A more recent survey by Liu et al. [12] provided a comprehensive review article that delved into the methodologies and strategies employed in pinpointing unfavorable events within potable water systems. The paper analyzed the current techniques, highlighting their strengths and limitations. The review is a pivotal resource for researchers and policymakers keen on improving water safety standards and management practices.

The above models did not address one central question: the time delay issue. The time delay parameter defines how long the model should wait before it triggers an alarm. Setting the delay time shorter than optimal will generate excessive false positives. Selecting a delay longer than optimal will generate extreme false negatives. It may also cause true positive alarms to be triggered with a delay, causing a loss of significant time, which is critical regarding water networks.

What is also not addressed explicitly by the above methods is the issue of true negative events (TNE). The literature uses the phrase true negative to refer to a situation in which the model did not trigger an alarm, and this action is correct. However, most records do not generate a warning in a typical dataset. This causes the calculation of Kappa statistics to be useless. This paper suggests that a TNE is a situation in which an event detected by the algorithm ended before it matured sufficiently to generate a trigger; i.e., it finished before the end of the delay period. For such a case, the TNE is correct. Hence, the TNE is essential to the model's efficiency calculation.

The tuning process of a WQM is considered a time-consuming process that can be performed most of the time by experts only. If performed poorly, the result may be either false positives or negatives. In other cases, it will lead to frustration and aborting the method. In both cases, the model turns out to be useless. Asking the end user, i.e., the water quality engineer, to allocate a substantial amount of time to calibrate their models seems impossible. Thus, one of the significant subjects addressed by the current paper is creating a simple tuning method.

Hence, the present paper addresses three issues:

- A method for selecting the correct time delay.
- Addressing the TNE.
- Describing a graphical calibration method enables the end user to calibrate the model easily.

The paper starts by describing a multiparameter method for water quality measurement. It follows what is described by Brill [13] and Brill and Brill [14]. Unlike [15] which used KMean based on three parameters, and thus assumed normal distribution of the quality measurements, the presented methodology is a more general one and uses two tuning parameters: DT – Delay time and PV. Given the above scenario, this paper then presents how each event type (TP – True Positive, FP – False Positive, TN – True Negative, FN – False Negative) is defined. Lastly, a graphical calibration method for tuning the algorithm parameters is presented. The second part of the paper implements the technique over a dataset from a monitoring station. The algorithm assigns a number to each of the events in the dataset. Then, the algorithm examines each event in a two-phase plane of DT and PV parameters. Given its location in this plane, a decision about whether to notify is made. In parallel, human experts classify the events as True or False events. By combining information from these two sources (Notify/Not Notify and True/False), a simple graphical demonstration of the data enables the user to select the thresholds for the model. Thresholds are the minimum level of DT (henceforth, DT<sub>min</sub>) and the minimum level of PV (henceforth, PV<sub>min</sub>). An illustrative example supports the implementation of the method.

## 2. Description and Calibration of the Proposed Model

This section describes the calibration of the WQM. Assume there is an online system in which online sensors measure the water quality. The online system records every fixed

interval (e.g., every minute or two). Each data record contains a timestamp and a vector of K measurements. The full dataset includes a  $K \times M$  table in which each of the K columns contains all the data measurements of a single sensor, and each row of the M rows contains K measurements for a specific timestamp.

Step 1:

The first step of the algorithm is to normalize the values in the dataset. Normalization refers to a process in which a data value will replace each value in the dataset from 0 to 1. This normalized data will allow determining the Euclidean distance (or any other metric distance method) over the complete set of variables without referring to the original physical units of each variable. Since the control system produces a new record every new time interval, we assume the dataset contains M rows at a certain point. Suppose that each row has an index m (1 to M). The algorithm calculates the normalized value of measurement k in row m by Equation (1):

$$\hat{v}_m^k = \frac{X_m^k - X_{min}^k}{X_{max}^k - X_{min}^k} \tag{1}$$

The terms in Equation (1) are as follows:

- $\hat{v}_m^k$ —normalized value at row m and column k.
- $X_m^k$ —value at row m and column k.
- $X_{max}^k$ —maximum value in column k
- $X_{min}^k$ —minimum value in column k.

Hence, the values are normalized based on the range between each measurement’s minimum and maximum value and not based on the standard deviation, as in Cheng et al. [16]. Please note that the algorithm also uses a filter mechanism by trimming exceptional records from both ends of each parameter axis, i.e., eliminating upper and lower single percentages from each column’s data before calculating the max and min values.

The pros and cons of normalizing values using the above method are beyond the scope of this paper. However, it is worth noting that an alternative normalized method based on the standard deviation may produce some bias in central measurements such as average, standard deviation, and others in the case when the distribution is highly non-symmetric, while the range-based method as implemented in this case is more immune to random bias in data.

Step 2:

The second step of the algorithm is to track the movement of the location of a Virtual Center Point (VCP) in an M-dimensional hyperspace. Figure 1 shows the traveling path of such a virtual point in a two-dimensional space. The normalized values are the VCP coordination at each timestamp (as calculated in step 1).

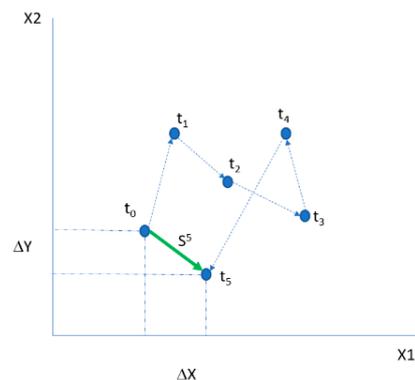


Figure 1. Actual Traveling Distance (ATD) of the virtual center point (VCP).

The VCP traveled from timestamp  $t_0$  to its location  $t_5$  within five timestamps. The term “total traveling distance” refers to the distance the VCP traveled from point  $t_0$  to point  $t_1$  until it arrived at  $t_5$ . The green arrow shows the actual traveling distance (ATD). Equation (2) shows the ATD between two records in the dataset with a time lag (LAG) of  $s$  timestamps, calculated using a standard Euclidean distance and given by Equation (2):

$$S^s = \sqrt{\sum_{k=1}^K (\hat{v}_k^t - \hat{v}_k^{t-s})^2}. \tag{2}$$

The two  $v$  terms in Equation (2) (inside the parentheses) refer to the normalized vectors. The first is for the last record, and the second is for the first location (located  $s$  steps back). In Figure 1, this means the data record at  $t_0$  and the data record at  $t_5$ . The algorithm calculates the term  $S^s$  for each record in the dataset. The number of data points recorded between the first and last in each calculation is called the LAG.

Step 3:

Figure 2a,b explain step 3 of the algorithm. The horizontal axis in each figure depicts the period: each tick in the axis is a different record (row) in the dataset. The vertical axis in each figure represents the distance traveled during a period that ends at this timestamp and has started  $s$  timestamps previously. This period refers to the LAG selected for the calculations, in this example, five steps. The green line in each figure is the corresponding curve for the distance traveled during the time stamp, ending in each time stamp. The difference between Figure 2a,b is the LAG size (in terms of records) used to calculate the actual traveling distance. The number of steps in Figure 2a is smaller than in Figure 2b. Hence, on average, the distance traveled by the virtual point under the Figure 2a regime is smaller than that traveled under the Figure 2b regime.

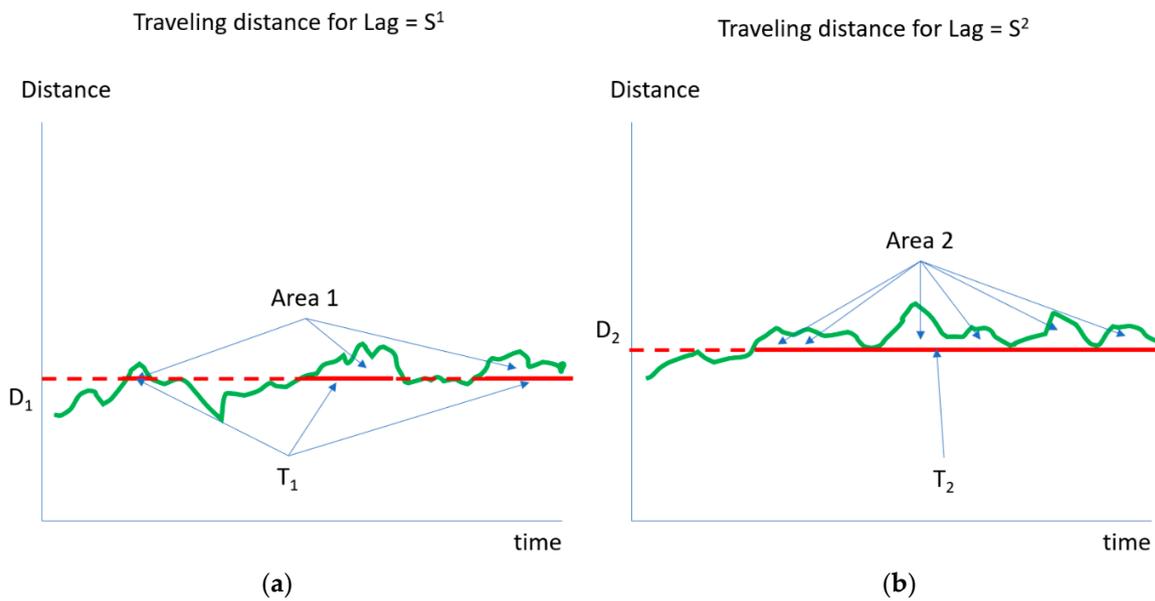


Figure 2. Relation between the Distance and Delay Time.

The red line in each of the figures is a different threshold. The term  $D_1$  shows the level of the red line in Figure 2a. The  $T_1$  value summarizes the cumulative time the green line value is above the red.

Figure 2b illustrates the exact mechanism. The higher  $D_2$  is, the smaller  $T_2$  is, and vice versa. Also, in Figure 2b, a negative relationship is obtained between  $D_2$  and  $T_2$ .

Each portion of an area between the red and green lines has a base. Figure 3 shows this base. Its units are minutes. Therefore, for each level of the red line represented by the Distance axis, it is possible to calculate the total time spent above the line by summing all the bases.

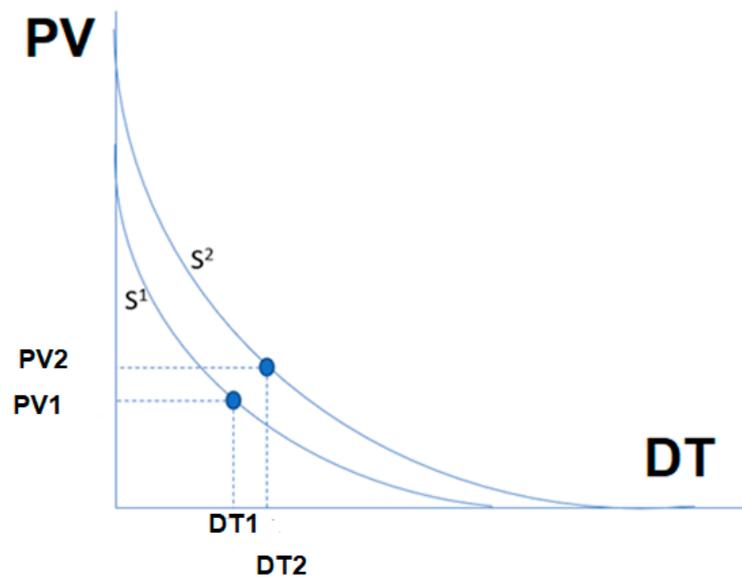


Figure 3. PV vs. DT for different S curves.

The discussion refers to Figure 3, which transforms the discussion into the control arena.

Speaking control-wise, it is assumed that when a process is characterized by high variability, it is recommended to wait only a short delay time before the alarm is declared. The lower the process variability, the higher the delay time before triggering an alarm. Process variability is expressed in this case as the ATD of the VCP as shown in Figure 1. Two different examples of the relations between the variability and the delay time are shown in Figure 3 as curves  $S^1$  and  $S^2$ . Each of these curves is a set of combinations of the Variability and Delay time for different LAG.

It is important to note that the  $S^1$  and  $S^2$  curves in Figure 3 are unknown. The process of discovering these curves (or at least one) is explained in what follows.

It is notable that the time scale of Figure 2 (the vertical axis) is not shown in Figure 3. The Vertical axis of Figure 3 depicts the PV. On the other hand, the delay time shown in Figure 2 on the vertical axis is shown in Figure 3 on the horizontal axis and is named DT. The points  $DT_1$  and  $PV_1$  in Figure 3, located on curve  $S^1$ , express a single combination of the delay time (expressed as the red line in Figure 2a) and variability as expressed by the green line in Figure 2a. Similarly, the point  $DT_2$  and  $PV_2$  in Figure 3, located on the curve  $S^2$ , expresses a single combination of delay time (expressed as the red line in Figure 2b) and variability, as expressed by the green line in Figure 2b.

Step 4

The stage is ready for introducing the tuning process based on the information presented thus far. The following discussion pertains to Figure 4, which illustrates a singular Delay ( $DT_0$ ) to Process Variability ( $PV_0$ ) curve.

We assume a specific S curve is selected (by choosing the lag time  $k$ ). A point on the selected S curve is selected (denoted by the green dot). This point corresponds to  $DT_0$  and  $PV_0$ , the delay time and the corresponding variability. Hence, any event in which the delay time is greater than  $DT_0$  and the distance traveled is more extended than  $PV_0$  will generate an alarm. Examples of such events are shown in points 1 and 3 in Figure 4. Any circumstance not fulfilling these two conditions will not create an alarm event. Examples of such a point are 2 and 4.

Additionally, suppose a human expert has classified all events as True (marked with a square) or False (marked with a triangle) based on process knowledge.

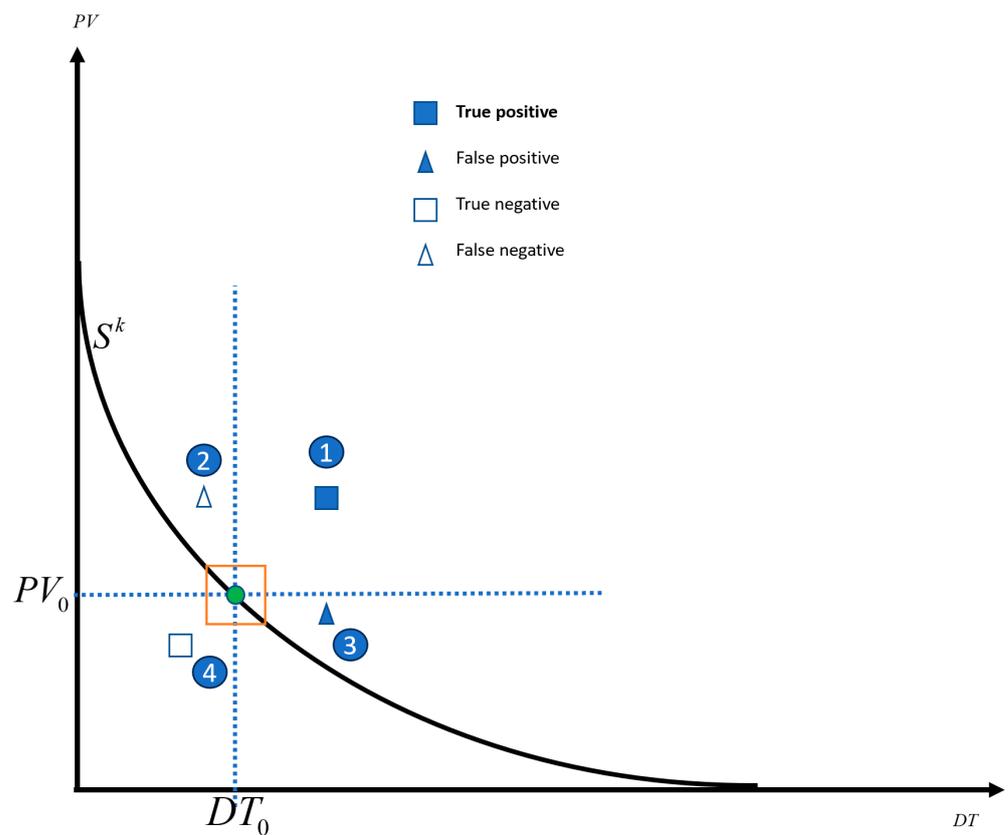


Figure 4. Events over decision thresholds.

Considering the information above, by merging the expert classification and algorithm classifications, each event can be categorized as one of the following:

- TRUE Positive events: Marked by filled squares. See point 1 as an example.
- FALSE Positive events: Marked by filled triangles. See point 4 as an example.
- TRUE Negative events: Marked by non-filled triangles. See point 2 as an example.
- FALSE Negative events: Marked by non-filled squares. See point 3 as an example.

The classification should refer to two facts: the event’s location in the Variability to Time Delay space relative to the green point and the expert’s subjective estimate concerning whether the specific event should or should not have generated an alarm. For example, point 1 in Figure 4 depicts an event that caused an alarm and was classified as True by the user. Hence, it will be classified as True Positive. Point 2 in Figure 4 mistakenly did not generate an alarm and was classified as a False Negative. Point 3 in Figure 4 mistakenly caused an alarm and is classified as a False Positive. Moreover, finally, point 4 in Figure 4, which depicts an event that did not create an alarm correctly, is classified as a True Negative.

Once a set of events is classified, as shown in Figure 4, a Kappa statistic can be evaluated for these points.

Step 5:

The calibration process has now reached its final stage—the estimation of parameters for the S curve. Assuming that the S curve has the functional form

$$PV = C + \alpha TD^b, \tag{3}$$

the task is to find a set of values for  $C, a, b$  including the green point and to generate the optimal Kappa.

Hence, the corresponding parameters are estimated using the least squares method for a given set of  $DT$  and  $PV$  points as measured for a given LAG  $S$ .

### 3. A Numerical Example

This section presents a numerical example that implements the above framework based on real-world data. Data from a water monitoring station comprise 5741 records. Each record contains the following information: Timestamp (one-minute interval between data records), Chlorine, Turbidity, pH, Conductivity, and Temperature. Figures 5–9 show charts of these five water quality variables.

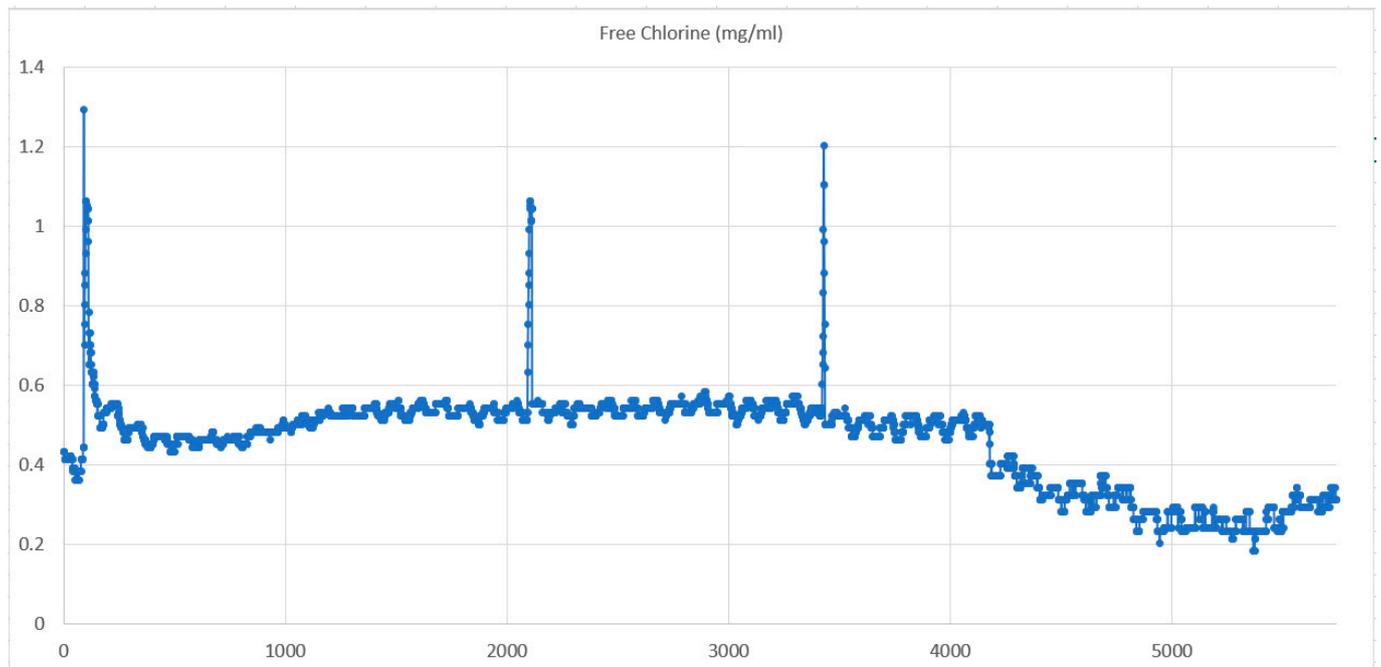


Figure 5. Raw data water quality chart for Chlorine (mg/mL).

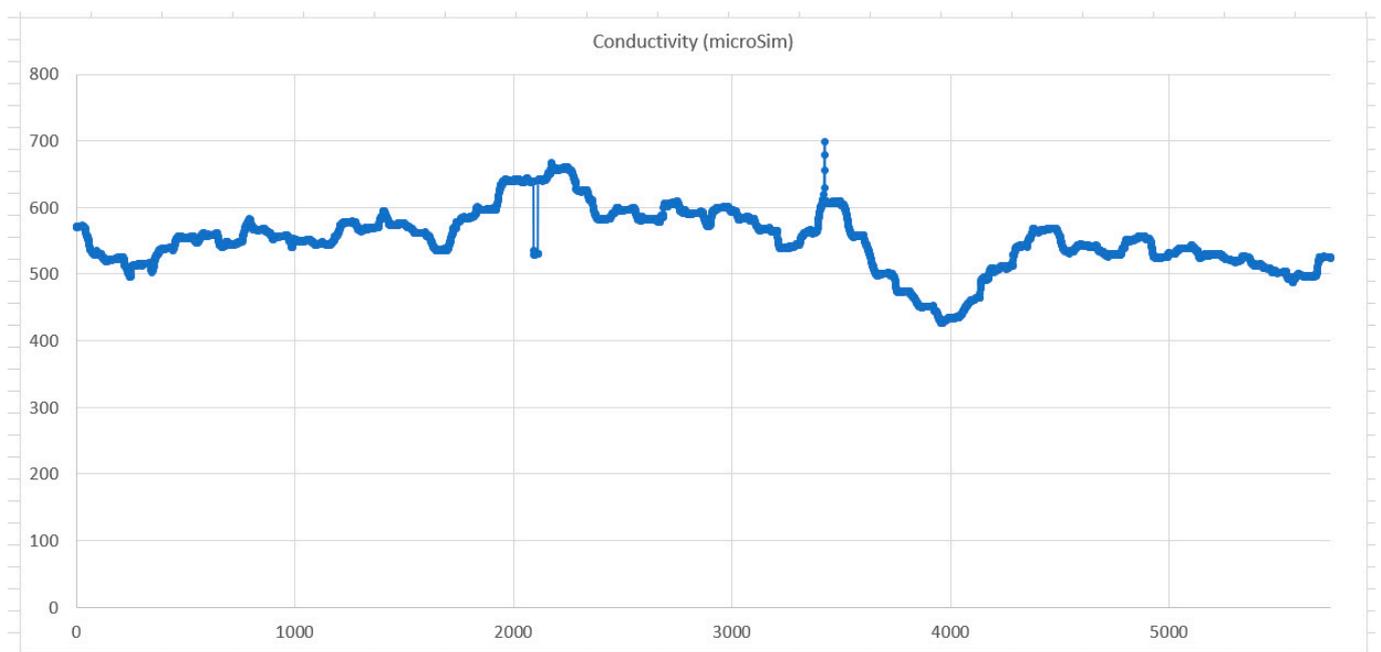


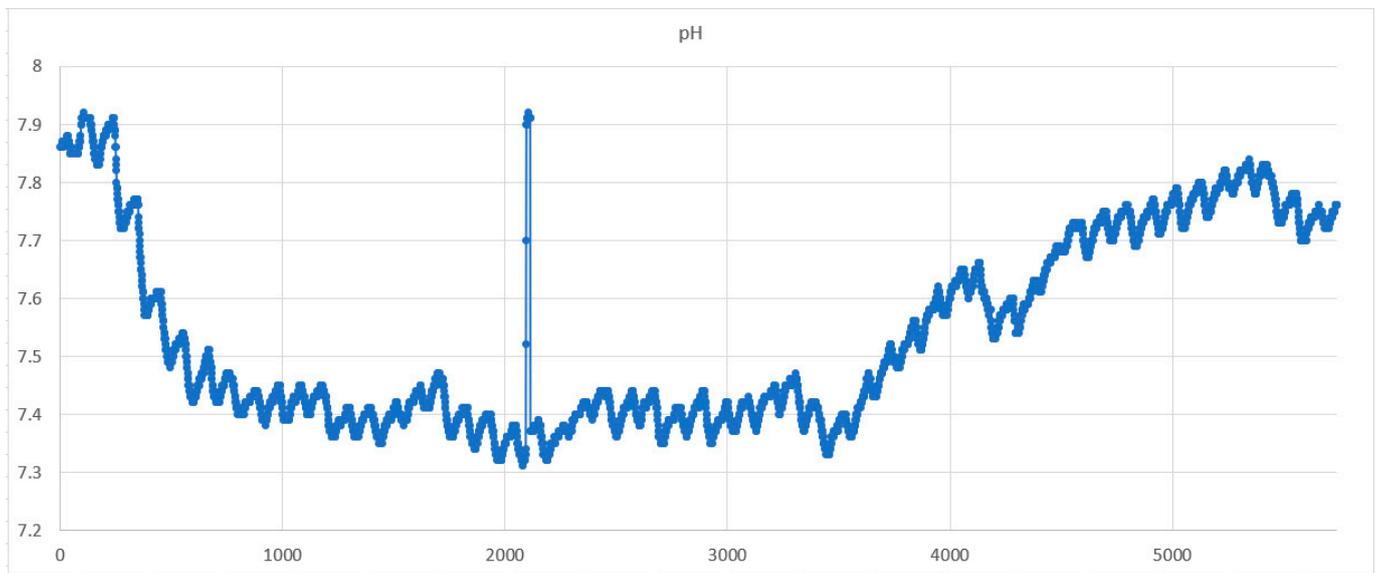
Figure 6. Raw data water quality chart for Conductivity (micro-Siemens).

Table 1 illustrates the principal attributes of the variable. The normalizing procedure uses these values to compute the traveling distance.

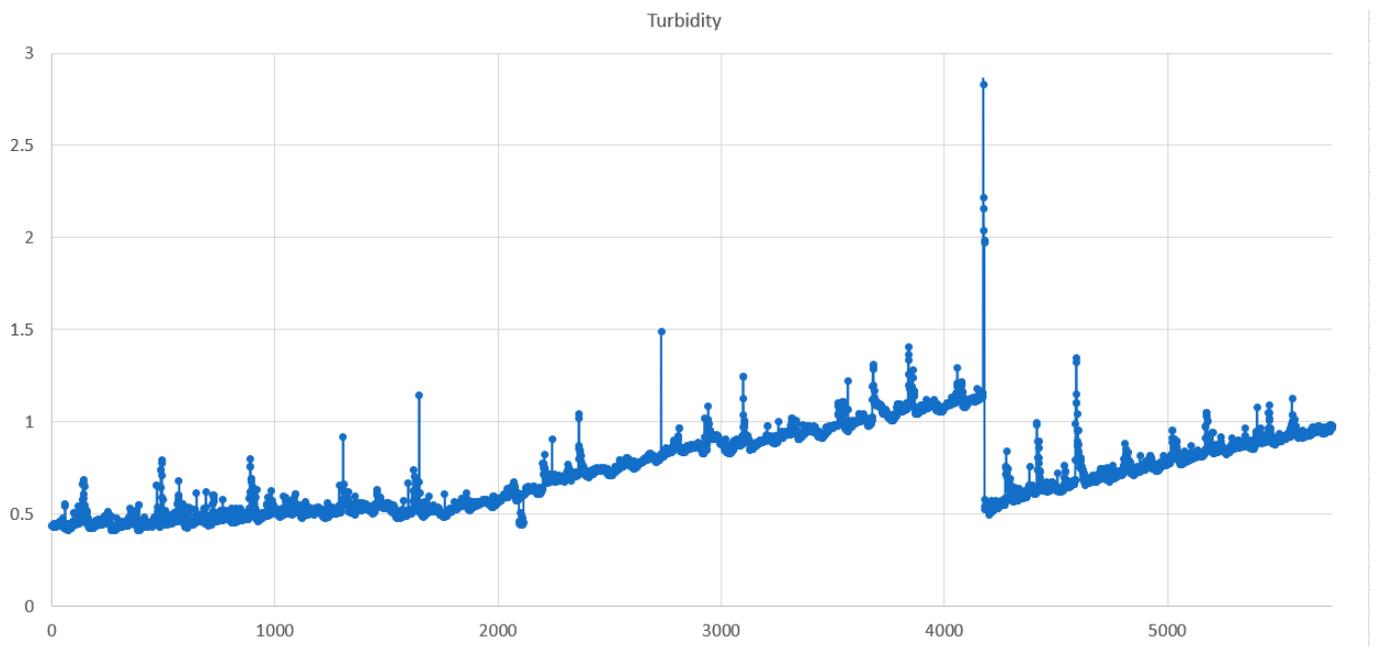
**Table 1.** Main attributes of the quality variables.

	CL	CO	pH	TU	TE
Max	1.29	699	7.92	2.832	31.9
Min	0.18	426	7.31	0.41	28
Median	0.5	552	7.45	0.715	29.9
Average	0.46	551.56	7.54	0.72	29.96
Std	0.12	45.20	0.17	0.21	0.95

By utilizing the information presented in Table 1 and Equations (1) and (2), the algorithm calculated the VCP’s traveling distance. Figure 10 shows the movement of the VCP using a lag time of 10 min and a red line of 0.08.



**Figure 7.** Raw data water quality chart for pH.



**Figure 8.** Raw data water quality chart for Turbidity.

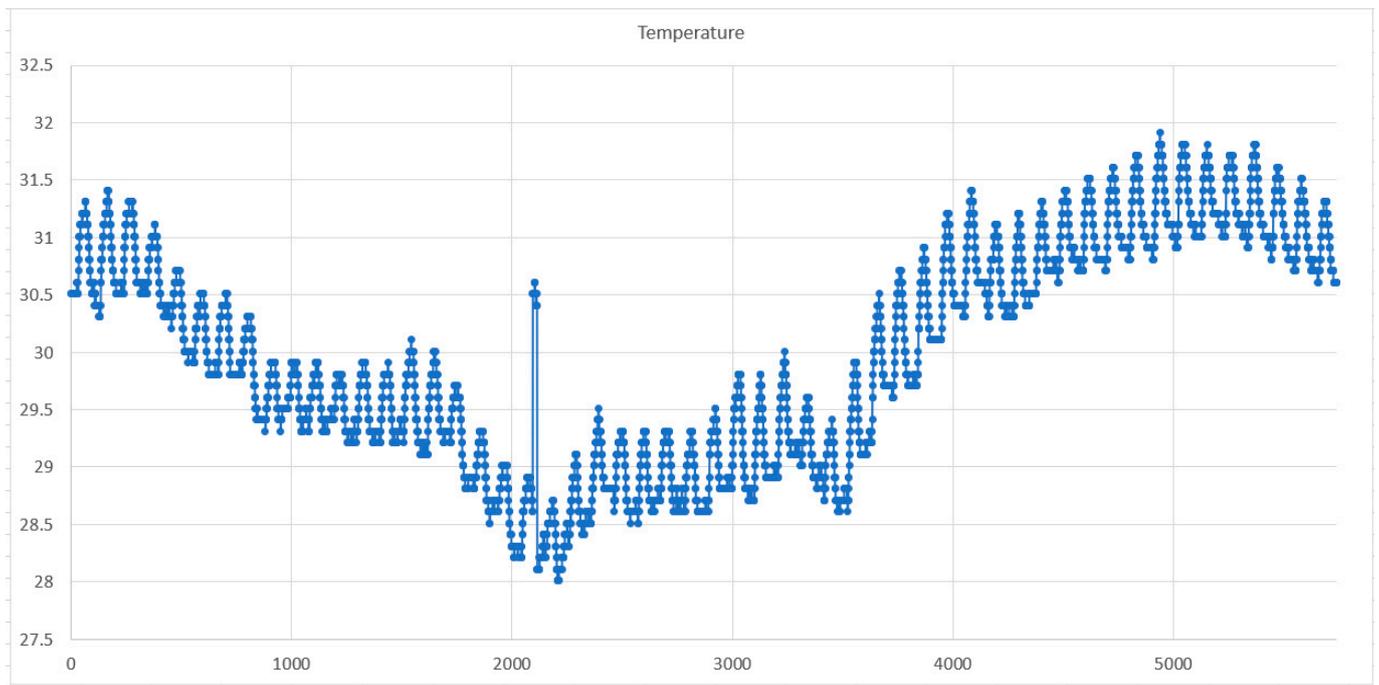


Figure 9. Raw data water quality chart for Temperature.

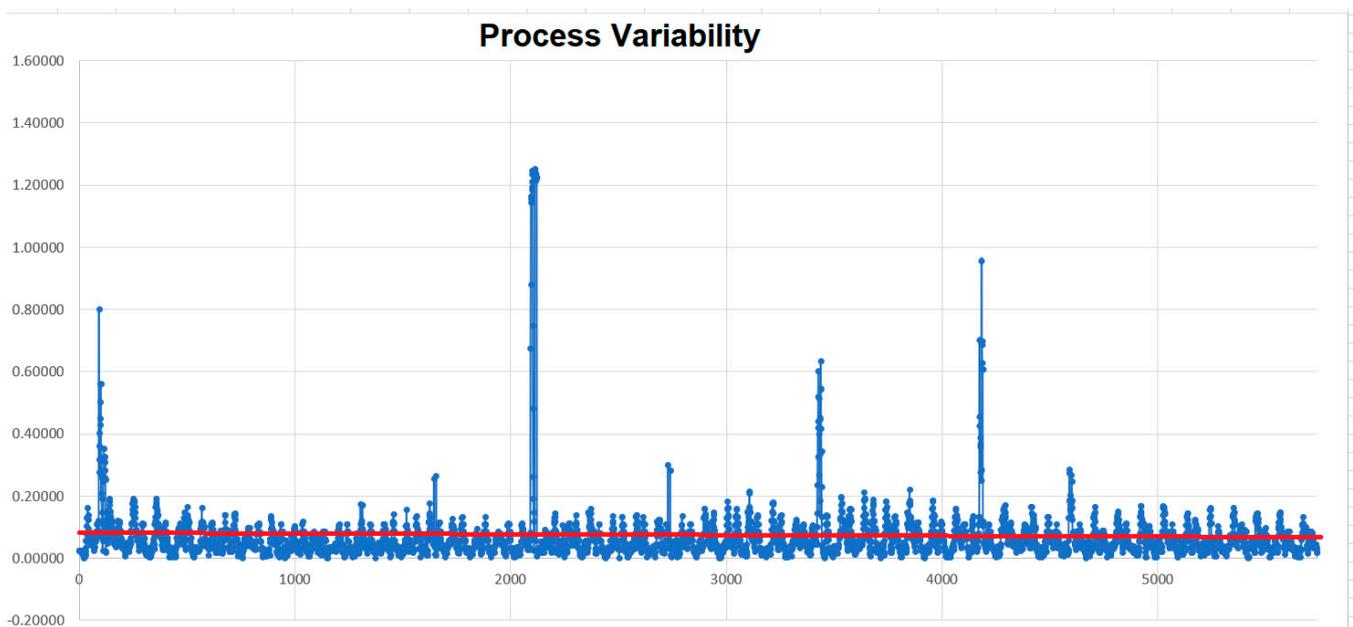


Figure 10. Process variability (PV) values using specific LAG and Threshold.

A Java program (see Supplementary Materials) was used to process the complete data set, including two events classified as positive events by an expert user. The processing procedure includes examining combinations of three nested loops.

Loop 1: Lag intervals between 2 and 15 min.

Loop 2: (nested to loop 1): Redline threshold between 0.1 and 3.0 with increment steps of 0.01.

Loop 3: (next to loop 2): Delay time between 5 and 15 min with intervals of 1 min.

The three nested loops yielded 3903 combinations. The algorithm processed the complete dataset for each combination and calculated the amount of TP, FP, TN, FN and the resulting Kappa statistics. The results of the best 20 combinations are listed in Table 2.

**Table 2.** Simulation results.

No	LAG	DC	Delay	Kappa	TN	TP	FN	FP	Total
1	13	0.26	7	0.764706	13	2	0	1	16
2	11	0.26	7	0.761905	12	2	0	1	15
3	14	0.25	9	0.761905	12	2	0	1	15
4	14	0.25	10	0.761905	12	2	0	1	15
5	14	0.25	11	0.761905	12	2	0	1	15
6	14	0.26	7	0.761905	12	2	0	1	15
7	14	0.26	8	0.761905	12	2	0	1	15
8	14	0.26	9	0.761905	12	2	0	1	15
9	14	0.26	10	0.761905	12	2	0	1	15
10	14	0.26	11	0.761905	12	2	0	1	15
11	8	0.25	7	0.758621	11	2	0	1	14
12	12	0.25	9	0.758621	11	2	0	1	14
13	13	0.25	9	0.758621	11	2	0	1	14
14	13	0.25	10	0.758621	11	2	0	1	14
15	8	0.26	7	0.754717	10	2	0	1	13
16	5	0.35	5	0.695652	4	2	0	1	7
17	5	0.36	5	0.695652	4	2	0	1	7
18	5	0.37	5	0.695652	4	2	0	1	7
19	5	0.38	5	0.695652	4	2	0	1	7
20	5	0.39	5	0.695652	4	2	0	1	7

Columns 2, 3, and 4 in the above table lists each combination's Lag time, DC, and delay time. Columns 5–9 lists the cases (TP, TN, FP, FN) and the corresponding Kappa. The top right column shows the total number of cases.

As seen from Table 2, the best combination is a LAG of 13 records, a delay time of 7 min, and a red threshold of 0.26. This combination is the best since it yields the best Kappa quickly, a critical issue in water quality detection.

#### 4. Conclusions

The current paper demonstrates a model calibration methodology that aims to detect abnormalities in water quality systems. This paper presented a graphical method for calibrating a WQM. According to the methodology proposed, the user may set three decision parameters: Lag time, Delay time, and Process variability expressed as Euclidean distance. Once these parameters have been set, a fully automated process (implemented as a Java program) can calculate the Kappa value. The user may choose the optimal Kappa based on the values for the LAG, Delay, and Variability. While calibrating, the user can also determine the probability of events for which they wish to receive notifications. Additional extensions of the above model could include the variable selection and lag time between records.

As seen in Figures 5–9 it may not be trivial for a user to set the optimal parameters based on the raw charts. The suggested method gives a straightforward methodology implemented in a Java code to solve this problem.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/computation11100200/s1>.

**Author Contributions:** Conceptualization, E.B. and M.B.; methodology, E.B. and M.B.; software, E.B.; validation, E.B.; formal analysis, E.B.; resources, E.B.; data curation, B.E writing—original draft preparation, E.B.; writing—review and editing, B.E and B.M; visualization, E.B. and M.B.; supervision, E.B.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** See Supplementary files.

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

## Glossary

Abbreviation	Term	Definition
PV	Process Variability	The amount of change in water quality for a given time window.
WQM	Water Quality Model	A model that answers the question if the water is usable.
DT	Delay Time	The time between the abnormality occurrence and alarm declaration
VCP	Virtual Center Point	The center of gravity of all measurements.
ATD	Actual Traveling Distance	The distance the VCP actually travels during LAG.
LAG	Time LAG	The amount of time before declaring an alarm.

## References

1. ISO/TS 24522:2019; Event Detection Process: Guidelines for Water and Wastewater Utilities. ISO: Geneva, Switzerland, 2019.
2. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. *Ann. Math. Stat.* **1967**, *38*, 325–339. [[CrossRef](#)]
3. Perelman, L.; Arad, J.; Housh, M.; Ostfeld, A. Event Detection in Water Distribution Systems from Multivariate Water Quality Time Series. *Environ. Sci. Technol.* **2012**, *46*, 8212–8219. [[CrossRef](#)] [[PubMed](#)]
4. Hou, D.; He, H.; Huang, P.; Zhang, G.; Loaiciga, H. Detection of water-quality contamination events based on multi-sensor fusion using an extended Dempster–Shafer method. *Meas. Sci. Technol.* **2013**, *24*, 055801. [[CrossRef](#)]
5. Liu, C.J.; Qiao, C.L. Water quality monitoring method based on data fusion technology. *Model. Meas. Control C* **2017**, *78*, 71–82. [[CrossRef](#)]
6. Hagar, J.; Murray, R.; Haxton, T.; Hall, J.; McKenna, S. The Potential for Using the CANARY Event Detection Software to Enhance Security and Improve Water Quality. In Proceedings of the World Environmental and Water Resources Congress, Cincinnati, OH, USA, 19–23 May 2013.
7. Hou, D.; Liu, S.; Zhang, J.; Chen, F.; Huang, P.; Zhang, G. Online Monitoring of Water-Quality Anomaly in Water Distribution Systems Based on Probabilistic Principal Component Analysis by UV-Vis Absorption Spectroscopy. *J. Spectrosc.* **2014**, *2014*, 150636. [[CrossRef](#)]
8. Nejari, F.; P'erez, R.; Puig, V.; Quevedo, J.; Sarrate, R.; Cuguer, M.A.; Sanz, G.; Mirats, J.M. Abnormal quality detection and isolation in water distribution networks using simulation models. *Water Eng. Sci.* **2012**, *5*, 67–72. [[CrossRef](#)]
9. Mao, Y.; Qi, H.; Ping, P.; Li, X. Contamination Event Detection with Multivariate Time-Series Data in Agricultural Water Monitoring. *Sensors* **2017**, *17*, 2806. [[CrossRef](#)] [[PubMed](#)]
10. García, D.; Creus, R.; Minoves, M.; Pardo, X.; Quevedo, J.; Puig, V. Data analytics methodology for monitoring quality sensors and events in the Barcelona drinking water network. *J. Hydroinform.* **2017**, *19*, 123–137. [[CrossRef](#)]
11. 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](#)]
12. Liu, X.; Lu, D.; Zhang, A.; Liu, Q.; Jiang, G. Data-driven machine learning in environmental pollution: Gains and prolems. *Environ. Sci. Technol.* **2022**, *56*, 11874–11887. [[CrossRef](#)] [[PubMed](#)]
13. Brill, E. Dynamic Brownian Motion with Density Superposition for Abnormality Detection. U.S. Patent 14/601,862, 9 April 2015.
14. Brill, E.; Brill, B. Identifying Water Network Anomalies Using Multi Parameters Random Walk—Theory and Practice. In *Waterbody Hydrodynamic Studies*; Daniel Malcangio, D., Ed.; InTech Open Science: London, UK, 2018.
15. Vergina, S.A.; Kayalvizhi, S.; Bhavadharini, R.M.; Kalpana Devi, S. A Real-Time Water Quality Monitoring Using Machine Learning Algorithm. *Eur. J. Mol. Clin. Med.* **2020**, *7*, 2035–2041.
16. Cheng, H.; Tan, P.-N.; Potter, C.; Klooster, S. Detection and Characterization of Anomalies in Multivariate Time Series. In Proceedings of the 2009 SIAM International Conference on Data Mining, Sparks, NV, USA, 30 April–2 May 2009.

**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.