Application and Optimization of Algorithms for Pressure Wave Evaluation Based on Measurement Data

Kai Vahldiek 1,*, Yao Yao 2, Bernd Rüger 3 and Frank Klawonn 1

1 Institute for Information Engineering, Ostfalia University of Applied Sciences, Salzdahlumer Str. 46/48, 38302 Wolfenbuttel, Germany; f.klawonn@ostfalia.de
2 Department of Computer Science, RWTH Aachen University, 52070 Aachen, Germany; yao.yao2@rwth-aachen.de
3 SWM Services GmbH, Emmy-Noether-Str. 2, 80992 Munchen, Germany; rueger.bernd@swm.de
* Correspondence: k.vahldiek@ostfalia.de; Tel.: +49-5331-939-31470

Featured Application: This work can be applied to find the pressure drop time points (PDTPs) for leakage localization in district heating networks (DHNs). It can be used to evaluate the pressure data of each sensor in the case of a large, spontaneous leakage. The PDTPs can be used within a framework to localize the leakage. It highlights the usage of different algorithms to find the PDTPs and presents the most suitable algorithm for several real events. It is strictly applied in DHNs where leakages are a demanding task to ensure an uninterrupted supply of energy.

Abstract: Leakages can occur in a district heating network, resulting in high economical damage. The propagating pressure wave resulting from large, spontaneous leakages reaches sensors at different locations in the network. This leads to pressure drops registered at each sensor at a different point in time. The time differences help to localize the leakage. Different algorithms are presented and applied in this paper to estimate the pressure drop time points based on non-uniform, time-discrete sensor signals. Five of the nine algorithms are self-developed with, e.g., parts of linear regression, whereas the other four algorithms have already been described in the literature, such as change-point detection. In this paper, several recorded events were investigated, and the algorithms were applied to real measurement data. After detection, leakage localization was performed to determine the affected exclusion area. A performance criterion was used as a measure to compare the algorithms. For practical application, the best-performing algorithm was identified. Furthermore, the events were classified according to how well they could be evaluated.

Keywords: district heating network; leakage localization; data analysis; linear regression; change-point detection; measurement data

1. Introduction

With increasing digitalization, more and more data are available. Today, not only systems and plants are digitally monitored, but also entire spatially large networks. Machines are digitally connected, and a large amount of data is also available about them. This data can be evaluated by applying various algorithms. Different sensors collect a wide range of various data. For individual machines, sensors are used to measure the condition of the machine. In plants or supply networks, important parameters within the process are measured. Typical types of sensor measurements are pressure, temperature, vibration, and light.

District heating networks (DHN), as part of a sustainable energy system, are eligible for a future renewable-energy-based heat supply [1]. Figure 1 shows a schematic presentation of a DHN.
Figure 1. Schematic representation of a district heating network.

DHNs, as a special form of thermo-hydraulic network, are used to supply customers with energy. The whole system is closed, pressurized, and operated with deionized water (a special, treated form of water) as a medium. Supply pipes and mass flow are drawn in red and return in blue. With a heat exchanger, the consumers obtain the energy from the network. The pressure difference between supply and return drives the mass flow through the heat exchanger, and the supply temperature is reduced to the return temperature. To this end, the network operator has to ensure minimal pressure difference and supply temperature, and the customer has to cool down the medium to a maximal allowed return temperature. The pressure difference is regulated by the circulation pump. If it is too low, the circulation pump power is increased, and if it is too high, pump power is decreased. The pressure maintenance pump ensures a certain pressure level throughout the whole network. At the supply site, the cooled-down medium is heated up again by a heat exchanger. Several heat sources are possible. A power plant is depicted in Figure 1. When transporting the medium from the power plant to the consumer, no medium should be lost in order to ensure an uninterrupted supply. However, leakages can occur within the pipes. In the case of a large, spontaneous leakage, loss of the transport medium can be compensated for a certain period of time. The replenishment is performed via the refill mass flow (green), but the available quantity for this is limited. The refill is necessary to maintain the prevailing pressure in the entire network and to keep the medium liquid.

Pressure sensors are installed in the network, and the data are permanently recorded [2]. With a monitoring program, the data can also be used to assist operator actions in the case of certain events [3]. If a leakage occurs, it is associated both with the loss of transport medium and with an initially local change in pressure conditions (a pressure drop). Immediately after the occurrence, the resulting pressure change spreads throughout the entire network. As soon as the pressure change is noticed by the pressure sensors distributed in the network, the pressurizer control reacts and feeds additional medium into the network. At each pressure sensor, the pressure wave is registered at a different point in time. A leak-
age causes the pressure to decrease rapidly within a few seconds. Determining the correct pressure drop time point (PDTP) in order to localize the leakage is important because, if the loss is too large and lasts too long, the entire network must be shut down. Shutting down, repairing, and restarting the entire network has an influence on economy and reliability [4]. For localization, hardware- or software-based approaches are available, such as image processing with infrared sensors or different data-driven methods with machine learning methods [5]. Another application for image analysis is the detection of oil spills with the usage of remote sensing methods [6]. Figure 2 shows exemplarily one exclusion area (EA) and the topology of the investigated network.

![Exemplary exclusion area (blue 3) and simplified network topology.](image)

**Figure 2.** Exemplary exclusion area (blue 3) and simplified network topology.

On the left side, the blue area belongs to EA 3. Each EA can be separated from the rest of the network by remote control. Ideally, only the EA affected by the leakage is separated [7]. The blue EA 3 is adjacent to the blue EAs 2, 4, and 7. Hence, three different shafts are visible with remote-controlled, motor-driven valves and pressure sensors. On the right side, a simplified network topology is shown, which is considered in this work. There is one supplier feeding in three different lines, which are connected directly or with linking lines. Altogether, \( N_{EA} = 28 \) EAs are present including the supplier. The topology yields six meshes and should describe the connections between the EAs.

The aim of this paper is the application and evaluation of nine different algorithms to detect the PDTPs in pressure data of 22 real events. Five algorithms are self-developed, and four have already been presented in the literature. After the application of the algorithms and the determination of the PDTPs, these time points are used to localize the leakage. For further real events, leakage localization is performed, as well as other methods such as those described in [8] and [9]. The PDTPs are used in an established framework presented in [10] to generate a ranking of the EAs. Ideally, the EA affected by the leakage is ranked first, but how well this works depends on the PDTPs determined by the algorithms. A performance criterion (PC) presented in [10,11] is used to compare the algorithms. It can be calculated for any number of results. In the case of a single ranking, the following holds: if the EA affected by the leakage is ranked first, the PC value is 100.0%; otherwise, if it is ranked last, the PC value is 3.6% (\( = \frac{1}{N_{EA}} \times \frac{1}{25} \)). The goal is to find the best-performing algorithm over all the events and to classify which events are better evaluable. The performance of algorithms can depend on the parameters chosen for evaluation. In this paper, the focus
lies on the evaluation time frame that is selected. For all other parameters, standard or best-known values are selected and kept constant, if there are any. The evaluation time frame can be optimized for historical events, as the EA affected by the leakage is known. An evaluation time frame is selected where the affected EA is ranked highest. However, for application in the case of a future real event, an evaluation time frame has to be set before the affected EA is known. Finally, for each algorithm, a recommendation is given for the best evaluation time frame and the application of the most suitable algorithm.

The paper is divided into six sections. After the introduction, Section 2 presents some related work. In Section 3, the background and motivation are given. The algorithms are described in detail in Section 4. Then, the results of the estimated PDTPs are used to localize the leakage with a specific framework for several real events in Section 5 before Section 6 briefly concludes.

2. Related Work

The detection of a PDTP is comparable with the task of change-point detection (CPD). With CPD, in general, a special point within a time series should be found. For detecting such a point, several algorithms and methods are available, which are used for determining PDTPs. With respect to DHNs, to our knowledge no literature is present about the use of such detection methods. Nevertheless, we provided a brief overview on the literature that could be relevant within other contexts, and we applied some ideas from them and further algorithms to our problem.

In [12], many CPD methods were proposed to detect change-points in time series. The examined methods included both supervised and unsupervised algorithms. Areas of application were medical condition monitoring, climate change detection, speech recognition, and image analysis. CPD defined the problem to find abrupt changes in data when a property of the time series changed. This could be different states within the data. The presented survey described different methods and introduced several criteria to compare the algorithms.

Since a variety of CPD methods exist and they are mostly tested only on artificial data, these methods were applied to real datasets in [13]. The datasets consisted of different time series. Human experts assigned the appropriate change-point information to each dataset. Evaluation metrics were introduced to compare the different methods. A benchmark for the used algorithms was established and discussed.

Within a time series, especially for sensor data, missing values are possible. In [14], a new algorithm for change-point detection was described for when the observed data may contain missing values. Missing data is particularly problematic for the application of conventional methods. The proposed method solved these problems by modelling dynamic distribution based on the underlying data as close to a time-varying, low-dimensional subset. Specifically, streaming data were used to track an approximation and measure deviations from those approximations. By computing various statistics on these deviations, the aim was to identify when the underlying manifold changed significantly or unexpectedly. Experiments and simulations demonstrated the robustness of the new method.

Changes in the activity of gas sensors were addressed in [15]. The main challenge was the turbulent nature of gas dispersion and the sensor response dynamics. An algorithm to detect change-points was proposed and evaluated on individual gas sensors. In addition, a sensor selection algorithm was developed, and the detected change-points, with respect to sensor position, were evaluated.

Besides change-point detection, anomaly detection methods can be used to detect special points in time series. In [16], an overview of methods to detect anomalies in sensor systems was given. In such systems with many sensors, anomaly detection is a particularly difficult problem since a trade-off between computational power and accuracy must be made within a limited framework. Ideally, such a change should be registered as quickly as possible. Various methods, from conventional techniques (statistical methods, time series analysis) to data-driven techniques (supervised learning, deep learning), were explored.
The impacts of different architectural environments (e.g., cloud solution) on the sensor ecosystem were also shown.

Leakage detection in DHNs can also be seen as anomaly detection. Therefore, opportunities for the application of machine learning models were presented in [17]. Leakages could be localized by using supervised ML methods that were trained with infrared images of the underground pipes. Different machine learning algorithms such as random forest, support vector machine, or logistic regression (LR) were applied. However, such approaches were not only used for leakage localization; they could also be applied to the extraction of heat load patterns or to assess the condition and quality of the installed valves.

Wireless sensor systems collect a large amount of data. With this large amount of data, the problem of detecting interesting parts of the datasets as automatically as possible arises. In [18], an algorithm for the online detection of anomalies in the measurements of a sensor system was proposed. This detection was evaluated on real datasets to demonstrate the reliability. The algorithm used piecewise linear models of time series that were representative and robust. This online detection allowed the computing of such models in near-real time, the building of them without prior knowledge of anomaly types, and the comparing of different time series of sensor data efficiently.

As discussed in [19], decision makers from the petroleum industry should be able to quickly identify anomalies in order to take appropriate action. The correct detection of such anomalies serves to avoid, correct, or react to the associated situations. In this application context, heavy machines were monitored by hundreds of sensors each, which sent data at a high frequency for damage prevention. A segmentation algorithm was presented to efficiently detect anomalies on the machines. A benchmark was also made for comparison with other methods.

The detection of road traffic anomalies is a very important aspect presented in [20]. Traffic anomalies can occur due to various reasons, such as the malfunctioning of sensors that collect traffic information. Therefore, a technique based on a statistical model was proposed for anomaly detection that identified temporal outliers in road traffic. A linear regression model and a z-score model were two statistical models that were used in combination to detect temporal outliers. The proposed technique could be used to detect sensor failures or unusual traffic incidents.

One method to detect change-points in time series is kernel-based online change-point detection. The algorithm in [21] used a kernel-based method and showed reliable detection results for a variety of problems. Some aspects of the algorithm were modified to improve the detection performance. Theoretical analysis and numerical simulations were performed to confirm the improvement of the results and to be more efficient compared to a state-of-the-art method.

Another method was presented in [22], where a Bayesian method was used to detect change-points. It determined whether there were changes in the mean or covariance at an unknown time in a sequence of independent multivariate normal observations. Different versions of the intrinsic Bayes factor were used to detect change-points. The results were tested and discussed on both simulated and real datasets.

A holistic algorithm for localizing leakages in pipelines was presented in [23]. A method for detecting leakages in pipelines was proposed that was based on acoustic emission event characteristics and a Kolmogorov–Smirnov test (KS). To apply the method, a sliding window was used to examine the data. A pipeline leak indicator was also determined using the KS test. The experimental results showed that the developed indicator accurately distinguished the leakages without any prior information. Moreover, it was more suitable than traditional features such as mean, variance, and kurtosis.

The literature review shows that there are some useful components in every presented publication, which are also taken up in this paper. The detection of PDTPs for localizing leakages in a DHN is the same as CPD in other contexts, such as those described in [12–15]. The missing values and their handling in [14] were not available, but due to the parametrization of transmission technology, there were only non-uniform, time-discrete
sensor values available. In the case of leakage and the occurrence of a pressure wave, there is a similar main challenge to that in [15]. The machine learning approaches from [17] can also be used for leakage localization but will not be pursued in this paper. Furthermore, if the dynamics of a sensor response is equal, no pressure wave attenuation is available. Every sensor registers the pressure wave at a different time point based on its position in the network. Algorithms for CPD and algorithms for anomaly detection can be used to detect PDTPs. Statistical methods are also used in this paper like those in [16], as well as the requirement to identify a change within the data as quickly as possible. A cloud solution for the implementation of the best algorithm could be an asset. Similar to [19], it is important for decision makers to enable a correct and fast reaction. After detecting the PDTP and localizing the leakage, the correct EA should be remotely separated from the rest of the network. Some of the developed algorithms in this paper use the ideas of a piecewise linear model [18] and linear regression for outlier detection from [20]. Further algorithms use ideas of kernel-based detection from [21], the Bayesian method from [22], and the KS test from [23]. The applied algorithms use the same ideas and methods as in [18,20–23].

3. Background and Motivation

With detection, localization, and separation, Figure 3 shows three very generic but still relevant steps in the case of a large, spontaneous leakage where a network shutdown is imminent. The damaged network part has to be separated as fast as possible to secure network operation. For two of them, there are solutions that have proved themselves in practical operation. Leakage detection has been realized reliably by monitoring the refill mass flow in real time, as described in [7,24]. As shown in Figure 2, fast separation of the damaged network part is possible due to the installed EAs. Hence, the crucial step is the localization of the leakage to separate the right EA from the rest of the network considered here.

Figure 3. Emergency management steps in the case of a large spontaneous leakage with imminent shutdown of network [10].

Localizing the leakage position in a network is a demanding task. In this paper, the localization was performed by evaluating the negative pressure wave. In the case of a leakage, the pressure drops at the leakage position, and a negative pressure wave propagates through the entire network and reaches each sensor at a different time point. The first step was the detection of a leakage, which was conducted by evaluating the refill mass flow. If the replenishment quantity started to increase, this was the leakage detection time point (LDTP). This process was described in detail in [7]. Then, the localization of the pressure wave consisted of two steps. In the first step, the PDTP for each sensor was estimated with one of the algorithms when the pressure wave was registered by the sensor. In the second step, these PDTPs were used to attribute the leakage location to the EAs. Both steps were already implemented in a prototype application in [7] that ran 24/7 without human interference. The last step was the remote separation of the affected EA.

The focus of this paper was on the performance of algorithms to determine the PDTPs. Several events shown in Figure 3 were investigated in order to determine the PDTPs of
each sensor for localizing the leakage. In general, all nine algorithms were applied to the events to determine the PDTPs. Overall, the pressure curves belonging to one event were similar in shape but unique for each event; hence, the preselection of algorithms was difficult, as each of them had its advantages and disadvantages. As described in [11], the shape of the pressure wave, which is influenced by the superpositions of reflected portions of the pressure wave and portions travelling along different ways in a complex network, constituted an additional challenge to find the “right” PDTP. Table 1 displays an overview of the real events evaluated in this paper.

Table 1. Evaluation of 22 real events.

| #  | Date      | Begin | Quantity [m³/h] | End     | Quantity [m³/h] | Delta [m³/h] | Gradient [m³/h/min] |
|----|-----------|-------|-----------------|---------|-----------------|-------------|---------------------|
| 1  | 27.07.2017| 17:38:00 | 23.5            | 17:39:30 | 78.0            | 54.4        | 36.3                |
| 2  | 27.07.2017| 17:42:45 | 78.0            | 17:44:30 | 146.1           | 68.2        | 39.0                |
| 3  | 27.07.2017| 17:51:30 | 146.1           | 17:52:15 | 221.4           | 75.2        | 100.3               |
| 4  | 27.07.2017| 17:54:30 | 221.4           | 17:55:15 | 251.9           | 30.6        | 40.8                |
| 5  | 01.08.2017| 08:23:24 | 84.0            | 08:24:00 | 94.0            | 10.0        | 16.7                |
| 6  | 01.08.2017| 09:02:00 | 46.1            | 09:09:30 | 113.5           | 67.4        | 9.0                 |
| 7  | 01.08.2017| 13:49:26 | 83.0            | 13:49:55 | 100.0           | 17.0        | 35.2                |
| 8  | 26.06.2018| 13:31:46 | 29.0            | 13:32:35 | 50.0            | 21.0        | 25.7                |
| 9  | 26.06.2018| 13:33:54 | 50.0            | 13:35:45 | 100.0           | 50.0        | 27.0                |
| 10 | 26.06.2018| 13:36:34 | 100.0           | 13:36:55 | 124.0           | 24.0        | 68.6                |
| 11 | 26.06.2018| 13:37:17 | 124.0           | 13:37:53 | 158.0           | 34.0        | 56.7                |
| 12 | 25.07.2018| 06:53:26 | 60.0            | 06:55:52 | 140.0           | 80.0        | 32.9                |
| 13 | 09.05.2020| 13:14:11 | 70.8            | 13:15:13 | 231.7           | 160.8       | 155.6               |
| 14 | 03.09.2020| 14:34:49 | 6.0             | 14:41:30 | 76.9            | 70.9        | 10.6                |
| 15 | 17.09.2020| 17:46:19 | 27.0            | 17:48:53 | 195.6           | 168.6       | 65.7                |
| 16 | 25.09.2020| 10:58:13 | 21.5            | 10:59:20 | 158.4           | 136.9       | 122.6               |
| 17 | 29.09.2020| 09:45:23 | 9.4             | 09:45:30 | 421.3           | 411.9       | 3530.0              |
| 18 | 28.07.2021| 16:59:25 | 43.9            | 16:59:53 | 122.0           | 78.1        | 167.4               |
| 19 | 28.07.2021| 17:02:40 | 46.9            | 17:03:30 | 149.9           | 103.0       | 123.6               |
| 20 | 28.07.2021| 17:30:46 | 100.5           | 17:31:21 | 200.7           | 90.0        | 154.3               |
| 21 | 30.09.2021| 09:28:47 | 90.0            | 09:28:58 | 133.0           | 43.0        | 234.5               |
| 22 | 07.10.2021| 08:14:08 | 39.8            | 08:17:12 | 254.2           | 214.4       | 69.9                |

If a leakage occurs or in the case of a refill process after maintenance work, there is a distinct increase of the refill mass flow at first. Afterwards, the refill mass flow stays approximately constant. The table shows the points in time of the start and end of the distinct increase at the beginning of each event with the respective replenishment quantities before and afterwards. In the case of a refill process, there can be several steps where refill mass flow rises, which correspond to the stepwise opening of a valve. This is performed to prevent a shutdown of network operation due to pressure falling below a minimal value. Furthermore, the differences between the quantities at the beginning and at the end (delta) are presented, as well as the speed of increase (gradient). For an event, the time point of a leakage occurrence is determined by the first increase of smooth replenishment [7]. This works also in the case of stepwise valve opening. Based on Figure 3, the localization step was divided into the determination of the PDTP and the attribution to exclusion area(s). The better the determination of the PDTPs, the better the localization functions [10]. In the scope of the current paper, there was the investigation of the best algorithm to determine the PDTP overall events. The mentioned PC from [10] was also used. Furthermore, it was also determined how well the events could be evaluated over all the algorithms.

For the selection of the different algorithms, prior knowledge of the physical processes in the case of a leakage is helpful. Both the evaluation and determination of a PDTP use the pressure data from each sensor, not the data on the replenishment. Leakage causes a local pressure drop that propagates through the entire network. The pressure wave reaches each sensor at a different time depending on its position in the network. In this work, a part of
one of the largest DHNs in Europe with a total length of more than 900 km was considered. This network has a length of approximately 90 km and is operated by Stadtwerke München. Altogether 119 sensors are installed within the network. The present DHN has a small spatial expansion compared to pipelines. Therefore, pressure wave attenuation should be extremely small or equal to zero. Apart from this, the change of replenishment and pressure are strongly correlated in time.

Within the scope of this paper, the most suitable algorithm for determining the PDTP was found for practical application. The evaluation was performed in two ways. Either the found time points were compared with the time points that were calculated from the leakage location and the pressure wave propagation velocity, or the framework from [10] was applied, which determined a ranking based on the PDTPs. The PC was the measure how well the EA affected by the leakage was identified. Decisive for the practice were the complete evaluation steps. Therefore, the algorithms were assessed on the basis of the known PC. Thus, possible positive or negative interactions between the PDTPs determined by one algorithm and the attribution step were included in this case.

It should be noted that the assessment of the algorithms was conducted for a small number of real events. Thus, the results may change if further events become available. However, for practical reasons, this was the best effort that could be completed. In particular, data from a real DHN under operation was used. Up to now, the cost–benefit ratio for artificial leakages, for example, by filling a tank truck, was rated not suitable. Due to temperature, this could be only applied to the return line. Because the realization of such an experiment would take place on a road space, it would require complex safety measures due to high pressures and leakage rates. However, the statistical planning of such an experiment could be useful to have a limited number of well-selected events available.

4. Algorithms

Based on 22 real events where a refill process acted on the rest of the DHN as a leakage, the corresponding PDTPs were determined. Nine algorithms were available, which are described in detail. All the algorithms were implemented within open source R statistics software [25]. The first five algorithms were self-developed, whereas the other four algorithms were already existing. Table 2 shows the nine algorithms used here.

The objective of the algorithms was to find the PDTPs in an evaluation time frame that contained the LDTP. Due to the physical behavior of DHNs, pressures had to change close to the LDTP. PDTPs as an input for the attribution of the EA (cp. Figure 3) had to be determined as precisely as possible. This was demanding because the sensors distributed in the DHN delivered measurement data as non-uniform, time-discrete values. Furthermore, only 60% of the data points were closer than 4–5 s to each other, which was very high compared to the 250 ms presented in [2,10] as a requirement for uniform, time-discrete data. Within the evaluation time frame, for each sensor \(i\) a number of data points \(N_i\) was present. The algorithms were applied to each sensor \(i\) separately. For simplification, \(i\) was not used as an index for the following descriptions of the algorithms. The chosen CPD algorithms from the literature were employed in this paper, but within a different scope: the task was not to detect a change-point in an approximate time interval. However, the requirement was the precise determination of the same distinct point in the times series data of several sensors.

The first algorithm was the mean algorithm (MAN algorithm, cp. Figure 4, top left). First, the mean value of the measurements \(M_{i,a}\) before time point \(j\) with \(j \in \{1, \ldots, N - 1\}\) and the mean value \(M_{i,b}\) after time point \(j + 1\) were determined for all \(j\). The maximum difference \(Q_i\) between the mean values showed the largest change in the signal. At this point \(j\), the pressure then began to drop, and this was the PDTP. For sensor \(i\), Equations (1)–(3) were used:

\[
M_{i,a}(j) = \sum_{t=1}^{j} \frac{p_i(t_n)}{j}
\]
\[ M_{i,b}(j) = \sum_{n=j+1}^{N_{i}} p_{i}(t_n) \]  

\[ Q_{i}(j) = |M_{i,a}(j) - M_{i,b}(j)| \]

The second algorithm was the difference algorithm (DFN algorithm, cp. Figure 4, top right). With this algorithm, the maximum difference \( D_i \) of each pressure curve between the value of data point \((j-1)\) and the value of data point \((j+1)\) was searched. If the maximum absolute difference of each pressure curve was found, the value and time of each pressure signal could be determined. The data point with the maximum absolute difference \( D_i \) was the PDTP, and Equation (4) was applied:

\[ D_i(j) = p_i(t_{j+1}) - p_i(t_{j-1}) \]

The third algorithm was the prediction algorithm (PDN algorithm, cp. Figure 4, middle left). Two points were predicted and compared with the next real point, respectively. The data points \(a\) with \(p_{i,1}^{*}(t_j)\) and \(c\) with \(p_{i,2}^{*}(t_j)\) were predicted and compared with the real data \(p_i\), respectively. The data point \(a\) was predicted with a linear regression \(p_{i,1}^{*}\) of the previous four data points \(\{j-4, j-3, j-2, j-1\}\). The data point \(c\) was also predicted with
linear regression $p_{i+2}^*$, but using the four data points $\{j - 3, j - 2, j - 1, j\}$. The data points $b$ and $d$ were real data points with $p_i(t_j)$ and $p_i(t_{j+1})$, respectively. The absolute differences $D_{i,1}$ between predicted data point $a$ and real data point $b$ and $D_{i,2}$ between predicted data point $c$ and real data point $d$ were calculated according to Equations (5) and (6):

$$D_{i,1}(j) = p_i(t_j) - p_{i+1}^*(t_j)$$  \hspace{1cm} (5)$$

$$D_{i,2}(j) = p_i(t_{j+1}) - p_{i+2}^*(t_{j+1})$$  \hspace{1cm} (6)$$

Then, point $a$ with the smallest $j$ for which the following conditions (Equations (7)–(9)) held, was selected as the PDTP:

$$D_{i,1}(j) < 0$$ \hspace{1cm} (7)$$

$$D_{i,2}(j) < 0$$ \hspace{1cm} (8)$$

$$D_{i,1}(j) < D_{i,2}(j)$$ \hspace{1cm} (9)$$

The fourth algorithm was the linear algorithm (LNR algorithm, cp. Figure 4, middle right). A linear line was constructed from each point to every other point with $t_{n+1} > t_j$ and $p_i(t_{n+1}) < p_i(t_j)$. At each point $j$, the sum of the negative gradients $g_{i,j}(n)$ of the linear lines passing through this point was calculated. The PDTP was located at the point where the sum of all the gradients $G_i(j)$ was at its maximum. For the calculation, Equations (10)–(12) were applied:

$$g_{i,j}(n) = \frac{p_i(t_{n+1}) - p_i(t_j)}{t_{n+1} - t_j} \text{ for } p_i(t_{n+1}) < p_i(t_j)$$  \hspace{1cm} (10)$$

$$g_{i,j}(n) = 0 \text{ for } p_i(t_{n+1}) \geq p_i(t_j)$$  \hspace{1cm} (11)$$

$$G_i(j) = \sum_{n=j}^{N-1} |g_{i,j}(n)|$$  \hspace{1cm} (12)$$

The last self-developed algorithm was the moving linear regression algorithm (MLR algorithm, cp. Figure 4, bottom left) as number five. There were two main factors that influenced this algorithm. One of them was the location of the starting point, and the other one was the scaling factor $s$. The starting point $t_{\text{max}}$ for the algorithm was set to the maximum value $p_{i,\text{max}}$ in the pressure data because a pressure drop should be detected and not an increase. From the starting point $t_{\text{max}}$, two more data points, $t_{\text{max}+1}$ and $t_{\text{max}+2}$, were used for starting a moving linear regression $p_i^*$ with $\{t_{\text{max}}, t_{\text{max}+1}, t_{\text{max}+2}\}$. The permitted prediction interval for $p_i^*$ was defined by the standard deviation $\sigma_i^*$ of the used data points $\{t_{\text{max}}, t_{\text{max}+1}, t_{\text{max}+2}\}$ for linear regression. The interval was expanded or reduced by the scaling factor $s$ to optimize the prediction interval. The point $j$ was checked by the following conditions according to Equation (13):

$$|p_i^*(t_j) - p_i(t_j)| < p_i(t_j) + s \cdot \sigma_i^*$$  \hspace{1cm} (13)$$

If the checked point $j$ was within this interval, it was not the PDTP. In addition, this point was also considered for the moving linear regression, and the next data point $j + 1$ was evaluated (which was then the new data point $j$ to be checked). The moving linear regression $p_i^*$ was calculated for $\{t_{\text{max}}, t_{\text{max}+1}, t_{\text{max}+2}, \ldots, t_{j-1}\}$. If the checked point $j$ was not within the interval, this point could be the PDTP or an outlier. For an outlier, the next data point $j + 1$ had to be in the interval. In this case, both points were used for the moving linear regression (and $j + 2$ was the new data point $j$ to be checked). However, if the point $j + 1$ was also outside of the interval, then the point $j$ was the PDTP.

The sixth algorithm was change-point estimation by a probabilistically pruned objective via the Kolmogorov–Smirnov statistic (KSS algorithm) [26]. For applying this algorithm, the R package “ecp” was used. The package was developed to be able to perform analyses of multiple change-points with as few assumptions as possible. Furthermore, the package is suitable for both univariate and multivariate observations. Hierarchical estimation is based on either a divisive or agglomerative algorithm. For divisive estimation, the change-points
are identified sequentially via a bisection algorithm. The agglomerative algorithm estimates
the location of change-points by determining an optimal segmentation. Both algorithms can
detect any type of distributional change within the data. This is an advantage over many
conventional change-point algorithms, as they can only detect changes within marginal
distributions [27]. The algorithm used an algorithm for multiple change-point analysis that
used dynamic programming and pruning for segmentation. The method used approximate
nonparametric estimation of multiple change-points. The cp3o framework (Change-Point
Procedure via Pruned Objectives) was developed for this purpose [26]. This procedure
applied a pruning routine to reduce the search space and computational cost. Therefore,
eXisting goodness-of-fit methods were directly utilized. For the goodness-of-fit measure,
the Kolmogorov-Smirnov statistic was applied. The algorithm found the location of the
change-point, which was the PDTP in the present case.

The seventh algorithm was the kernel-based change-point detection algorithm (KCP
algorithm) [28]. The KCP algorithm was also implemented in the mentioned R package
“ecp” and is an algorithm for multiple change-point analysis that uses the ‘kernel trick’ and
dynamic programming for segmentation. The assumption of the multiple change-point
problem is that the data belonging to a set are equipped with a positive semidefinite kernel.
With a model-selection penalty, the selection of a number of change-points is allowed by
using the kernel-based detection method of Harchaoui and Cappe [28]. For univariate data,
the change-in-mean procedure was applied. The method found the location of the
change-point as a change-point within a single time series of each dataset.

The eighth algorithm was the change-point detection in mean (CPD algorithm) [29].
This algorithm used the package “changepoint” in R. With this package, different change-
point search methods can be selected and applied. The function we used from this package
identified changes in mean and variance, calculating the optimal positioning and, if nec-
essary, the number of change-points for data using the specified method. It was assumed
that the data were normally distributed. The used parameter ‘AMOC’ stands for at most
one change-point, which matched with the present data because only one change-point,
the PDTP, should be found [29]. With changes in the mean and variance of a dataset, the
change-point was found where the change was at its maximum. The same applied to
variance because the method detected simultaneous changes in mean and variance [30].
With this combination, it was possible to detect the change-point that was the PDTP.

The last algorithm was the Bayesian single change-point (BCP algorithm). For this
algorithm, the package “bcp” in R was used. In this package, a Bayesian analysis for
change-point problems is proposed, and the given methods from Wang and Emerson
are implemented [31]. A specific case was defined by the product partition model for
the normal-error change-points. For univariate (or multivariate) Bayesian change-point
analysis, it was assumed that the unknown partition of a time series into blocks existed
and the mean was kept constant within each block [31]. The time series was divided
into blocks where Bayesian methods were used to estimate the change-points or block
boundaries of the unknown partitions. The method also supported multivariate change
point problems [32]. The result was the estimated posterior probabilities of changes at each
location, where the location with the highest probability was the PDTP.

The pressure data were examined in a period of time of five minutes available. Around
the LDTP described in Section 3, pressure data 150 s before and after the time point were
evaluated. It was possible that each algorithm needed another time frame for application.
Therefore, the applied time frame could be optimized in retrospect for the most suitable
time frame. The selection of the time frame was conducted in such a way that the affected
EA was in the highest-possible place in the ranking. The time frame needed time borders
T1 (left side of the LDTP) and T2 (right side of the LDTP) that could be defined and then
optimized. However, when applying the algorithms to new pressure data of an event
with an unknown leakage location, no optimization of the evaluation time frame could
be performed. Therefore, the best-fitting time frame on average was selected for each
algorithm individually. The time frames were expected to be different for each algorithm.
There were algorithms that could benefit from more data before the LDTP or afterwards. Each algorithm had its own optimized time frame.

Table 2. Overview of the nine algorithms for determining the pressure drop time points.

| #   | Algorithm                                      | Development  |
|-----|-----------------------------------------------|--------------|
| 1   | Mean algorithm (MAN)                          | Self-developed|
| 2   | Difference algorithm (DFN)                    | Self-developed|
| 3   | Prediction algorithm (PDN)                    | Self-developed|
| 4   | Linear algorithm (LNR)                        | Self-developed|
| 5   | Moving linear regression algorithm (MLR)       | Self-developed|
| 6   | Change-point estimation by probabilistically pruned objective (via Kolmogorov–Smirnov statistic) (KSS) | Literature [26] |
| 7   | Kernel-based change-point detection algorithm (KCP) | Literature [28] |
| 8   | Change-point detection in mean (CPD)           | Literature [29] |
| 9   | Bayesian single change-point (BCP)            | Literature [31] |

5. Results and Discussion

The algorithms were applied to the 22 real events (cp. Table 1). The evaluation time frame in which the data were evaluated by an algorithm was determined by $T_1$ and $T_2$ in seconds. These were selected in the following ranges: $T_1 \in \{-120, -100, \ldots, -20\}$ s and $T_2 \in \{20, 40, \ldots, 120\}$ s. Each algorithm was applied to the 22 real events for each possible evaluation time frame $[T_1, T_2]$ to detect the PDTPs and localize the leakage of the event. The localization was performed with an established framework in the second evaluation step of Figure 3 described in [10]. In Figure 5, a heat map is presented for one exemplary algorithm (BCP) and the arbitrarily selected event 22.

| Event | $T_1$ | $T_2$ | PC value |
|-------|-------|-------|----------|
| 22    | 20    | 30    | 46.4     |
| 22    | -30   | 30    | 46.4     |
| 22    | -40   | 30    | 46.4     |
| 22    | -50   | 30    | 46.4     |
| 22    | -60   | 30    | 46.4     |
| 22    | -70   | 30    | 46.4     |
| 22    | -80   | 30    | 46.4     |
| 22    | -90   | 30    | 46.4     |
| 22    | -100  | 30    | 46.4     |
| 22    | -110  | 30    | 46.4     |
| 22    | -120  | 30    | 46.4     |

Figure 5. Heat map for the BCP algorithm for event 22 (green—max PC value, red—min PC value). Three PC values are highlighted: best (dashed black rectangle), worst (white rectangle), and the one for best overall evaluation time frame (black rectangle).

The heat map illustrates the evaluability of event 22 for the BCP algorithm. Both times, $T_1$ (rows) and $T_2$ (columns), were varied. For each evaluation time frame combination, the PDTPs were determined for the event, and the ranking for the affected EA was shown as the PC. Green values indicate higher PC values than red values. The maximum PC value of this event is dark green in a black dashed rectangle within the map. The optimal evaluation time frame for the BCP algorithm and event 22 was $[T_1, T_2] = [-40, 30]$. If multiple maximum PC values were available, the row sum and column sum were calculated. The evaluation time frame with the highest row sum and column sum for the maximum PC weight was determined.
value was selected to be the optimal evaluation time frame for this event. In contrast, the worst PC value is marked red in a white rectangle in the same way as in the case of multiple equal minimal values. Later, the best overall time frame for all the events was evaluated. In the case of the BCP algorithm, it was \([T_1, T_2] = [−100, 30]\). The corresponding PC value is highlighted by a black rectangle. Based on the heat maps (BCP algorithm for all events), Table 3 was created to show the optimal time frames and maximum PC values. In addition, the PC value for the latter determined the best overall time frame for all events, also shown for each single event.

Table 3. Optimal time frame and maximum PC value of each event for the BCP algorithm and PC values for best overall time frame.

| #  | T1   | T2   | PC    | PC \([T_1, T_2] = [−100, 30]\) |
|----|------|------|-------|-------------------------------|
| 1  | −110 | 70   | 96.4  | 35.7                          |
| 2  | −20  | 20   | 100.0 | 46.4                          |
| 3  | −110 | 20   | 100.0 | 85.7                          |
| 4  | −20  | 20   | 100.0 | 57.1                          |
| 5  | −90  | 50   | 100.0 | 100.0                         |
| 6  | −30  | 110  | 100.0 | 78.6                          |
| 7  | −120 | 70   | 96.4  | 60.7                          |
| 8  | −40  | 50   | 96.4  | 82.1                          |
| 9  | −100 | 60   | 100.0 | 100.0                         |
| 10 | −30  | 70   | 100.0 | 75.0                          |
| 11 | −60  | 30   | 100.0 | 100.0                         |
| 12 | −50  | 110  | 100.0 | 92.9                          |
| 13 | −90  | 80   | 100.0 | 100.0                         |
| 14 | −110 | 20   | 96.4  | 92.9                          |
| 15 | −30  | 40   | 100.0 | 85.7                          |
| 16 | −110 | 30   | 100.0 | 35.7                          |
| 17 | −70  | 120  | 89.3  | 85.7                          |
| 18 | −50  | 20   | 100.0 | 53.6                          |
| 19 | −120 | 30   | 92.9  | 92.9                          |
| 20 | −20  | 90   | 96.4  | 92.9                          |
| 21 | −100 | 20   | 100.0 | 50.0                          |
| 22 | −40  | 30   | 100.0 | 96.4                          |
| mean | −69  | 53   | 98.4  | 77.3                          |

Using the BCP algorithm, evaluating each event resulted in many different optimal time frames and very high individual PC values. A total of 15 out of the 22 real events were evaluated by the BCP algorithm with a PC value of 100.0%. In these cases, the affected EA was ranked in the first place. With this algorithm, the worst PC value was 89.3% for event 17. For practical application, it was not possible to optimize the evaluation time frame, as the affected EA was not known. Moreover, an evaluation time frame must be selected prior to application. Later, the best overall time frame for all events for each algorithm was determined. For the BCP algorithm, the best overall time frame was \([T_1, T_2] = [−100, 30]\). The last column shows the PC value for each event when choosing this best overall time frame. The affected EA was ranked in the first place four times with a PC value of 100.0%. The range of the PC values was 35.7 to 100.0%. Therefore, the evaluability of the events with the BCP algorithm was very different and correct localization was not always possible using the best overall time frame. The drop in the mean PC from 98.4 to 60.5% for the optimal and overall best evaluation time frames, respectively, indicated that practical application was difficult when judging the available measurement data for these 22 events. A detailed investigation should be performed to clarify reasons for that large drop and is out of the scope of this paper.

Table 4 shows the maximum PC value for all the algorithms for each event, no matter what combination of \(T_1\) and \(T_2\) led to that result. The optimal time frames are not shown for reasons of simplicity. The algorithms in the table are sorted in descending order based
on the mean maximum PC value in the last row. A PC value of 100.0% indicates a first place in the ranking. The KSS, CPD, and MAN algorithms had the highest PC values on average with 98.5%. The DFN algorithm was the worst with a PC value on average of 89.8%. This algorithm also had the worst maximum PC value of 39.3% for event 8. In general, the maximum PC value of 100.0% was reached 118 times. This means that an evaluation time frame \([T_1, T_2]\) existed for which the affected EAs were ranked in the first place in 60% of all cases. For each algorithm, the affected EA was in first place at least once for one event. For events 7, 17, and 20, no algorithm ranked the affected EA in the first place, yet.

Table 4. Maximum PC values of all algorithms for each event for arbitrary optimal \([T_1, T_2]\).

| #  | KSS  | CPD  | MAN  | BCP  | MLR  | KCP  | PDN  | LNR  | DFN  |
|----|------|------|------|------|------|------|------|------|------|
| 1  | 96.4 | 100.0| 96.4 | 96.4 | 92.9 | 100.0| 100.0| 92.9 | 100.0|
| 2  | 100.0| 100.0| 96.4 | 100.0| 96.4 | 92.9 | 92.9 | 96.4 | 92.9 |
| 3  | 100.0| 96.4 | 100.0| 100.0| 96.4 | 92.9 | 100.0| 57.1 | 100.0|
| 4  | 100.0| 96.4 | 100.0| 100.0| 100.0| 100.0| 85.7 | 96.4 | 100.0|
| 5  | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 6  | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 7  | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 92.9 |
| 8  | 100.0| 100.0| 100.0| 100.0| 96.4 | 96.4 | 96.4 | 96.4 | 92.9 |
| 9  | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 67.9 |
| 10 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 96.4 |
| 11 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 12 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 13 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 14 | 100.0| 96.4 | 100.0| 100.0| 96.4 | 96.4 | 100.0| 82.1 | 100.0|
| 15 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 71.4 |
| 16 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 17 | 85.7 | 92.9 | 92.9 | 89.3 | 92.9 | 96.4 | 96.4 | 96.4 | 92.9 |
| 18 | 96.4 | 96.4 | 96.4 | 100.0| 96.4 | 96.4 | 96.4 | 96.4 | 92.9 |
| 19 | 96.4 | 96.4 | 92.9 | 92.9 | 100.0| 96.4 | 96.4 | 92.9 | 96.4 |
| 20 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 96.4 | 92.9 | 64.3 |
| 21 | 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0| 100.0|
| 22 | 100.0| 100.0| 100.0| 100.0| 92.9 | 96.4 | 96.4 | 96.4 | 100.0|
| mean| 98.5 | 98.5 | 98.5 | 98.4 | 97.7 | 96.3 | 95.3 | 91.4 | 89.8 |

To find the overall best time frame for practical application for all the algorithms, a summary of the 198 (9 algorithms for 22 events) available heat maps was necessary to obtain a meaningful heat map (as in Figure 5) for each algorithm. All 22 heat maps of all the events for one algorithm were summarized. In Figure 6, the summarized PC values for each algorithm for all the events are also shown as a heat map. The heat map illustrates the summarized evaluability of all the events for each algorithm. All nine algorithms are presented row-wise (first row: BCP, CPD, DFN; second row KCP, KSS, LNR; third row MAN, MLR, PDN). The maximum summarized PC value is, again, dark green in a black box within each map. In contrast, the worst PC value is also red in a white box. Hence, the dark green PC value represents the maximum summarized PC value, and the overall best time frame could be determined directly.

It was evident that each algorithm had an individual overall best time frame, which is shown in Table 5. Algorithms BCP, CPD, and MAN, benefitted from more data points before LDTP \(T_1 = \{-100, -100, -90\}\) and from fewer after LDTP \(T_2 = \{30, 20, 30\}\). For algorithms DFN, LNR, and PDN, the opposite was beneficial. Fewer data points before LDTP \(T_1 = \{-60, -30, -30\}\) and more data points after LDTP \(T_2 = \{120, 120, 110\}\) led to larger PC values. Fewer data points before and after the LDTP were good for algorithms KCP, KSS, and MLR with \(T_1 = \{-40, -20, -50\}\) and \(T_2 = \{30, 20, 40\}\). The heat maps had a relatively similar range between the max and min values for the summarized PCs of the DFN and LNR algorithms. This also applied to the BCP and CPD algorithms.
Figure 6. Heat map of the summarized PC values for all events for each algorithm (green—max PC value, red—min PC value; coloring for each heat map separately).

Table 5. Mean and standard deviation, max and min PC, from heat map and overall best time frames.

| # | Algorithm | Mean | Standard Deviation | Max PC | Min PC | Overall Best [T1, T2] |
|---|-----------|------|--------------------|--------|--------|----------------------|
| 3 | PDN       | 61.2 | 4.0                | 70.6   | 51.8   | [−30, 110]           |
| 9 | BCP       | 60.5 | 5.7                | 77.3   | 46.3   | [−100, 30]           |
| 7 | KCP       | 55.8 | 6.0                | 66.9   | 33.3   | [−40, 30]            |
| 8 | CPD       | 54.5 | 8.8                | 71.3   | 31.0   | [−100, 20]           |
| 6 | KSS       | 53.7 | 6.1                | 71.6   | 39.8   | [−20, 20]            |
| 1 | MAN       | 53.3 | 5.0                | 66.7   | 36.7   | [−90, 30]            |
| 5 | MLR       | 51.5 | 5.4                | 68.0   | 39.6   | [−50, 40]            |
| 2 | DFN       | 48.6 | 6.1                | 60.1   | 33.4   | [−60, 120]           |
| 4 | LNR       | 48.4 | 6.3                | 60.9   | 31.5   | [−30, 120]           |

Table 5 shows the mean PC of the summarized heat maps and the standard deviation, as well as the max and min PC values. The table is sorted in descending order according to the mean PC value. The PDN algorithm had the highest mean PC value with 61.2%. In contrast, the LNR algorithm had the worst mean PC value with 48.4%. This was comparable to the maximal and minimal values, where the PDN algorithm had the highest values and the LNR algorithm also had the lowest PC values. Furthermore, the PDN algorithm showed the smallest standard deviation; hence, this algorithm was the most insensitive to
change in the time frame compared to the other algorithms. To this end, CPD was the most affected by a change in the evaluation time frame. The overall best evaluation time frame for each algorithm was the one belonging to the maximal PC value in the summarized heat map. In the case of the BCP algorithm, it was \([T_1, T_2] = [-100, 30]\). When calculating the mean time value, shown in Table 3, it resulted in a different evaluation time frame \([T_1, T_2] = [-69, 53]\). However, this selection led to worse results. Table 5 indicates that each algorithm benefitted from an individual best overall evaluation time frame \([T_1, T_2]\).

This is an important result for practical application because it illustrated which algorithm should be applied in which individual best overall time frame. Within this evaluation time frame, the algorithm was applied, and the pressure data were evaluated. After determining the overall best time frames, each algorithm was applied with its time frame to all 22 real events. The results are shown in Table 6.

Table 6. PC values of each algorithm for all events within the optimal time frame.

| #  | BCP | KSS | CPD | PDN | MLR | KCP | MAN | LNR | DFN | Mean |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 12 | 92.9| 57.1| 67.9| 89.3| 100.0| 78.6| 100.0| 96.4| 100.0| 86.9 |
| 15 | 85.7| 100.0| 60.7| 96.4| 100.0| 100.0| 82.1| 100.0| 32.1| 84.1 |
| 7  | 60.7| 92.9| 17.9| 92.9| 92.9| 92.9| 92.9| 85.7| 92.9| 80.2 |
| 10 | 75.0| 64.3| 92.9| 60.7| 85.7| 92.9| 89.3| 50.0| 96.4| 78.6 |
| 13 | 100.0| 100.0| 3.6| 82.1| 57.1| 64.3| 100.0| 82.1| 100.0| 76.6 |
| 11 | 100.0| 96.4| 3.6| 92.9| 96.4| 100.0| 3.6| 92.9| 100.0| 76.2 |
| 14 | 92.9| 100.0| 96.4| 75.0| 57.1| 28.6| 96.4| 100.0| 28.6| 75.0 |
| 3  | 85.7| 92.9| 96.4| 92.9| 82.1| 92.9| 17.9| 17.9| 82.1| 73.4 |
| 2  | 46.4| 92.9| 82.1| 53.6| 46.4| 89.3| 53.6| 96.4| 85.7| 71.8 |
| 6  | 78.6| 7.1| 100.0| 96.4| 67.9| 89.3| 32.1| 100.0| 71.4| 1.4  |
| 22 | 96.4| 46.4| 89.3| 14.3| 82.1| 10.7| 100.0| 96.4| 64.6| 71.8 |
| 19 | 92.9| 92.9| 92.9| 57.1| 96.4| 89.3| 7.1| 17.9| 67.9| 68.3 |
| 20 | 92.9| 92.9| 92.9| 32.1| 92.9| 28.6| 32.1| 32.1| 65.5|     |
| 4  | 57.1| 71.4| 50.0| 67.9| 100.0| 50.0| 51.7| 46.4| 67.9| 63.1 |
| 8  | 82.1| 100.0| 50.0| 100.0| 35.7| 100.0| 71.4| 7.1| 3.6| 61.1 |
| 17 | 85.7| 57.1| 39.3| 10.7| 67.9| 25.0| 85.7| 85.7| 78.6| 59.5 |
| 18 | 53.6| 71.4| 89.3| 92.9| 53.6| 50.0| 50.0| 35.7| 35.7| 59.1 |
| 5  | 100.0| 25.0| 92.9| 89.3| 46.4| 71.4| 85.7| 3.6| 10.7| 58.3 |
| 21 | 50.0| 71.4| 100.0| 100.0| 96.4| 7.1| 78.6| 7.1| 7.1| 57.5 |
| 16 | 35.7| 28.6| 50.0| 50.0| 7.1| 50.0| 100.0| 96.4| 50.0| 52.0 |
| 9  | 100.0| 25.0| 100.0| 10.7| 3.6| 3.6| 96.4| 3.6| 3.6| 38.5 |

Mean(cp. Table 5) 77.3 71.6 71.3 70.6 68.0 66.9 66.7 60.9 60.1
Mean(cp. Table 4) 98.4 98.5 98.5 95.3 97.7 96.3 98.5 91.4 89.8
Diff -21.1 -26.9 -27.3 -24.7 -29.7 -29.4 -31.8 -30.5 -29.7

The table represents the nine algorithms for the 22 real events. Each algorithm was applied in its own overall best evaluation time frame \([T_1, T_2]\). For all the events, the mean PC value was calculated, which corresponds to the max PC value in Table 5. In addition, the mean for an arbitrary optimal time frame with the highest possible PC value for the algorithms is presented (cp. Table 4). The last row contains the difference between these two values. The algorithms in the table are sorted in descending order based on the mean PC value. For each algorithm, the affected EA was in the first place at least once for one event. As already mentioned, the first place in the ranking led to a PC value of 100.0% and the last place to a PC value of 3.6%. For each event, every algorithm performed differently. Therefore, different algorithms should have been applied to different events, but this was not applicable in practice as well. Based on the mean value of each algorithm, the BCP algorithm performed best within its overall best time frame with a mean PC of 77.3%. This was larger than the 70% suggested as a minimal value to be ready for practical use [10]. For events 5, 9, 11, and 13, the affected EAs were ranked in the first place. To be on the first three places, a PC value of \(\geq 92.9\)% was necessary. The BCP algorithm ranked the affected EAs for
nine events (5, 9, 11, 12, 13, 14, 19, 20, and 22) in the first three places. For this comparison, the KSS and CPD algorithms were better because they ranked for ten events the affected EAs in the first three places. Both also had the affected EA in the first place for four events like the BCP algorithm. The worst algorithm within its own overall best time frame was the DFN algorithm with a mean PC value of 60.1%. When the differences between the PCs for arbitrary optimal and overall best time frames were compared, the BCP algorithm had the smallest difference with a value of −21.1%. Therefore, this algorithm had high PC values within both the overall best time frame and an arbitrary time frame. The MAN algorithm had the highest difference with a value of −31.8%.

Either all optimal time frames were similar to the overall best time frame or the BCP algorithm was the most robust against changes in the evaluation time frame. Based on its overall best time frame with more data points before LDTP ($T_1 = −100$) and only a few data points after LDTP ($T_2 = 30$), a pressure curve with a strong drop was identified. Therefore, the most important drop within the time frame was detected. As described in Section 4, the division of the pressure curve into multiple blocks and the use of Bayesian methods to estimate change points was an advantage of this algorithm. Larger differences within the pressure curve could be better determined within these blocks. In contrast, the DFN algorithm showed the worst mean PC values for the overall best time frame. This was due to the fact that many differences within the pressure curve could be identified as a maximum difference with the location of the PDTP. However, these differences could also be normal noise differences between different data points. Therefore, this algorithm was not that robust, and the low PC values indicated a wrong detection of the PDTP. Overall, all the algorithms needed their own overall best time frame to evaluate the pressure curves and determine the PDTPs. The BCP algorithm had the highest best mean PC value and the lowest difference; this algorithm should be implemented in a practical setting and used for the determination of the PDTPs. It should be noted that the number of events was limited and the data quality of events seemed to vary when judging the mean evaluability of each event. The rating of the algorithms might change if further data become available or insight into data quality issues is gained.

6. Conclusions

In the case of large spontaneous leakages in a DHN, a shutdown is imminent. Fast separation of damaged network parts is important to secure the supply for customers. To this end, leakage localization is most important, as detection and separation can be handled well by monitoring the refill mass flow and using already available EAs. Leakage localization is a demanding task that can be tackled by different means. Here, evaluation of the negative pressure wave was considered. In previous publications, it has been shown that attribution of the EA works in a suitable way if the PDTP can be determined most precisely, such as in [8,10,11]. The scope of this paper was the determination of the PDTPs. There are several algorithms available to evaluate pressure sensor data. The aim was to assess the algorithms based on measurement data for all the events recorded in a real DHN to find the best-performing algorithms for practical application. As a novelty, self-developed algorithms and already existing algorithms normally used for other applications were applied here for leakage localization. Further, the influence of the time frame used for evaluation was investigated. It should be noted that the assessment of the algorithms was conducted for a small number of events of a real DHN in operation. Thus, the results can change if further events become available. Nevertheless, the achieved results were the best that are currently possible, especially as conducting experiments is difficult, expensive, and connected to certain risks.

Nine different algorithms were evaluated and applied to pressure data of twenty-two real events. Five of them were self-developed, and four were selected from the literature. The algorithms were explained in detail and then applied to 22 real events. Instead of rating the PDTPs found by the algorithms directly, they were used in the second evaluation step for the attribution of EAs (cp. Figure 3). Possible positive or negative interactions
between the PDTPs and attribution were included, and the influence of the algorithms on total leakage localization was assessed. For localization of the leakage, the PDTPs were used in an established framework presented in [10] to generate a ranking of the EAs. To find the most suitable algorithm overall for the events, a PC in the range of 0 to 100% was used to compare the algorithms.

In the case of each event, refill flow rose in a time range of several seconds up to minutes, and the change covered 10 to 400 m$^3$/h with gradients from 9 to 3530 m$^3$/h. Altogether, the events were unique. The LDTP was set to be at the very beginning of the refill flow rise. Data at a maximum of 120 s before and after the LDTP were used for determining the PDTP by the algorithms. First, all possible evaluation time frames $[T_1, T_2]$ were evaluated, and the PC was determined for each algorithm and all the events. It was shown that the evaluation time frame $[T_1, T_2]$ had a significant influence on the PC. In 15 out of 22 events, there was an optimal time frame that resulted in a PC of 100% in the case of the BCP algorithm. The KSS, CPD, and MAN algorithms had the highest PC values on average, with 98.5% for the optimal time frames. The DFN algorithm was the worst with a PC value on average of 89.8%. Furthermore, with 60%, the affected EA was in the highest rank for all events and all algorithms in the case of optimal time frames. The affected EA was in the first place at least once for one event and all algorithms. However, for a few events, no algorithm ranked the affected EA in the first place.

The time frame could be optimized for historical events, as the affected EAs were known. For future events, it was not possible to use optimal time frames for practical application, as the affected EA is not known. Hence, the time frame had to be set before applying the algorithms. Thus, the overall best time frame for each algorithm was determined. Therefore, the PC values of each algorithm for all the events and all possible time frames were summarized and illustrated in nine heat maps in Figure 6. Each algorithm had an individual overall best time frame $[T_1, T_2]$. Furthermore, the mean PC value was calculated for all the possible time frames. The PDN algorithm had the highest mean PC value with 61.2%. In contrast, the LNR algorithm had the worst mean PC value with 48.4%. The PDN algorithm showed the smallest standard deviation. This algorithm was the most insensitive to change in the time frame compared to the other algorithms. To this end, CPD was the most affected by change in the evaluation time frame.

After the determination of the overall best time frame $[T_1, T_2]$ for each algorithm, they were applied with their individual overall best time frames to all 22 real events. The affected EA was in the first place at least once for one event and algorithm. Hence, every algorithm achieved another PC for an event, and multiple algorithms were necessary to determine the correct PDTP and localize the leakage. However, this is not applicable in practice, as well. As mentioned before, the optimization of the time frame and selection of the best-performing algorithm were only possible in cases of known leakages. Based on the mean value of each algorithm, the BCP algorithm was the most suitable within its overall best time frame with a PC of 77.3%. This algorithm should be applied in practice, and the overall best time frame should be set to be $[T_1, T_2] = [-100, 30]$. In 9 of 22 real events, the affected EAs were ranked in the first three places. The BCP algorithm was selected because it had the highest PC value on average. Based on its used Bayesian methods to estimate change-points, it is also suitable if the pressure data are changing.

Future work will focus on improved algorithms to determine PDTPs from measurement data. Since each event is unique and further real events are unknown, the chosen algorithm should be maximally robust. For now, the chosen BCP algorithm will be implemented in practice with its overall best time frame. In the case of the next real event, the algorithm will be able to show its full potential. Regarding events, a more detailed investigation should be performed, as totally different evaluability levels, on average, were found in the range between 38.5 and 86.9%. Furthermore, the large drop in PC values when changing from the optimal to the overall best time frame should also be investigated. As a selection of algorithms based on real events, ranking might change with new insights, and the described evaluation of the algorithms can be reconducted in the case of new events.
Author Contributions: Conceptualization, K.V. and B.R.; Methodology, K.V. and Y.Y.; Software, K.V.; Validation, B.R.; Formal Analysis, F.K.; Investigation, K.V. and Y.Y.; Resources, B.R.; Data Curation, K.V.; Writing—Original Draft Preparation, K.V.; Writing—Review and Editing, B.R. and F.K.; Visualization, K.V.; Supervision, F.K.; Project Administration, B.R.; Funding Acquisition, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Federal Ministry of Economic Affairs and Climate Action (BMWK), grant numbers 03ET1624 B and C.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the Federal Ministry of Economic Affairs and Climate Action (BMWK) and the Project Management Juelich (PtJ) for funding this research and development project as a part of the Federal Government’s 6th Energy Research Programme (funding code: 03ET1624 B and C).

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

References

1. Lund, H.; Østergaard, P.A.; Nielsen, T.B.; Werner, S.; Thorsen, J.E.; Gudmundsson, O.; Arabkoohsar, A.; Vad Mathiesen, B. Perspectives on fourth and fifth generation district heating. *Energy* 2021, 227, 120520. [CrossRef]
2. Valincius, M.; Vaišnoras, M.; Kaliatka, A. Study and demonstration of pressure wave-based leak detection in a district heating network. *Struct. Infrastruct. Eng.* 2018, 14, 151–162. [CrossRef]
3. Villalobos, F.; Hay, S.; Weidlich, I. Monitoring in a District Heating Pipeline System. In *Energy Geotechnics*; Ferrari, A., Laloui, L., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 132–139.
4. Zhou, S.; O’Neill, Z.; O’Neill, C. A review of leakage detection methods for district heating networks. *Appl. Therm. Eng.* 2018, 137, 567–574. [CrossRef]
5. Buffa, S.; Fouladfar, M.H.; Franchini, G.; Lozano Gabarre, I.; Andrés Chicote, M. Advanced Control and Fault Detection Strategies for District Heating and Cooling Systems—A Review. *Appl. Sci.* 2021, 11, 455. [CrossRef]
6. Tysiąc, P.; Strelets, T.; Tuszyńska, W. The Application of Satellite Image Analysis in Oil Spill Detection. *Appl. Sci.* 2022, 12, 4016. [CrossRef]
7. Rüger, B.; Pierl, D.; Guber, M.; Yin, J.; Baur, M.; Eberhard, H.; Klawonn, F.; Michels, K. Online Leakage Attribution to Exclusion Areas Prototype Application. *Energy Procedia* 2018, 149, 575–584. [CrossRef]
8. Vahldiek, K.; Koltermann, J.; Pierl, D.; Rüger, B.; Nürnberger, A.; Michels, K.; Klawonn, F. Leakage Localization in District Heating Networks Based on Real Network and Measurement Data. *Energy Rep.* 2021, 7, 508–516. [CrossRef]
9. Pierl, D.; Vahldiek, K.; Geisler, J.; Rüger, B.; Michels, K.; Klawonn, F.; Nürnberger, A. Online model- and data-based leakage localization in district heating networks—Impact of random measurement errors. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 2331–2338. [CrossRef]
10. Vahldiek, K.; Rüger, B.; Klawonn, F. Optimal sensor placement and influence of noise on pressure wave evaluation for leakage localization in a district heating network. *Sustain. Energy Grids Netw.* 2022, 30, 100672. [CrossRef]
11. Vahldiek, K.; Rüger, B.; Pierl, D.; Klawonn, F. Optimal Sensor Placement for Pressure Wave Detection for Leakage Localization in a District Heating Network. *Energy Rep.* 2021, 7, 499–507. [CrossRef]
12. Aminikhanghahi, S.; Cook, D.J. A Survey of Methods for Time Series Change Point Detection. *Knowl. Inf. Syst.* 2017, 51, 339–367. [CrossRef]
13. Van den Burg, G.J.J.; Williams, C.K.I. An Evaluation of Change Point Detection Algorithms. *arXiv* 2020, arXiv:2003.06222v3.
14. Xie, Y.; Huang, J.; Willett, R. Change-Point Detection for High-Dimensional Time Series with Missing Data. *IEEE J. Sel. Top. Signal Process.* 2013, 7, 12–27. [CrossRef]
15. Pashami, S.; Lilienthal, A.J.; Trincavelli, M. Detecting changes of a distant gas source with an array of MOX gas sensors. *Sensors* 2012, 12, 16404–16419. [CrossRef]
16. Erhan, L.; Ndubuaku, M.; Di Mauro, M.; Song, W.; Chen, M.; Fortino, G.; Bagdasar, O.; Liotta, A. Smart anomaly detection in sensor systems: A multi-perspective review. *Inf. Fusion* 2021, 67, 64–79. [CrossRef]
17. Mbijydenuyuy, G.; Nowaczyk, S.; Knutsson, H.; Vanhoupt, D.; Brage, J.; Callíkus, E. Opportunities for Machine Learning in District Heating. *Appl. Sci.* 2021, 11, 6112. [CrossRef]
18. Yao, Y.; Sharma, A.; Golubchik, L.; Govindan, R. Online anomaly detection for sensor systems: A simple and efficient approach. *Perform. Eval.* 2010, 67, 1059–1075. [CrossRef]
19. Martí, L.; Sanchez-Pi, N.; Molina, J.M.; Garcia, A.C.B. Anomaly detection based on sensor data in petroleum industry applications. *Sensors* 2015, 15, 2774–2797. [CrossRef]
20. Mondal, M.A.; Rehena, Z. Road Traffic Outlier Detection Technique based on Linear Regression. *Procedia Comput. Sci.* **2020**, *171*, 2547–2555. [CrossRef]

21. Bouchikhi, I.; Ferrari, A.; Richard, C.; Bourrier, A.; Bernot, M. Kernel Based Online Change Point Detection. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, 2–6 September 2019; pp. 1–5. [CrossRef]

22. Son, Y.S.; Kim, S.W. Bayesian single change point detection in a sequence of multivariate normal observations. *Statistics* **2005**, *39*, 373–387. [CrossRef]

23. Rai, A.; Ahmad, Z.; Hasan, M.J.; Kim, J.-M. A Novel Pipeline Leak Detection Technique Based on Acoustic Emission Features and Two-Sample Kolmogorov-Smirnov Test. *Sensors* **2021**, *21*, 8247. [CrossRef]

24. Filippini, E.; Marini, I.; Ongari, M.; Pedretti, E. District heating leakage measurement: Development of methods. *Energy Procedia* **2018**, *149*, 297–306. [CrossRef]

25. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2021.

26. Zhang, W.; James, N.; Matteson, D. Pruning and Nonparametric Multiple Change Point Detection. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 288–295. [CrossRef]

27. James, N.A.; Matteson, D.S. ecp An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data. *J. Stat. Soft.* **2014**, *62*, 1–25. [CrossRef]

28. Arlot, S.; Celisse, A.; Harchaoui, Z. A Kernel Multiple Change-point Algorithm via Model Selection. *arXiv* **2012**, arXiv:1202.3878.

29. Killick, R.; Eckley, I.A. changepoint An R Package for Changepoint Analysis. *J. Stat. Soft.* **2014**, *58*, 1–19. [CrossRef]

30. Chen, J.; Gupta, A.K. (Eds.) Univariate Normal Model. In *Parametric Statistical Change Point Analysis*; Birkhäuser Boston: Boston, MA, USA, 2012; pp. 7–88.

31. Erdman, C.; Emerson, J.W. bcp An R Package for Performing a Bayesian Analysis of Change Point Problems. *J. Stat. Soft.* **2007**, *23*, 1–13. [CrossRef]

32. Wang, X.; Emerson, J.W. Bayesian Change Point Analysis of Linear Models on Graphs. *arXiv* **2015**, arXiv:1509.00817.