Identifying Missing Sensor Values in IoT Time Series Data: A Weight-Based Extension of Similarity Measures for Smart Manufacturing*

Alexander Schultheis[1,2], Lukas Malburg[1,2], Joscha Grüger[1,2], Justin Weich[1,2], Yannis Bertrand[3], Ralph Bergmann[1,2], and Estefanía Serral Asensio[3]

[1] German Research Center for Artificial Intelligence (DFKI) Branch Trier University, Behringstraße 21, 54296 Trier, Germany
[2] Artificial Intelligence and Intelligent Information Systems, Trier University, 54296 Trier, Germany, https://www.wi2.uni-trier.de
[3] Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Warmoesberg 26, 1000 Brussels, Belgium

{alexander.schultheis,lukas.malburg,joscha.grueger, justin.weich,ralph.bergmann}@dfki.de
{schultheis,malburgl,grueger,s4juweic,bergmann}@uni-trier.de
{yannis.bertrand,estefania.serralasensio}@kuleuven.be

Abstract Smart Manufacturing integrates methods of Artificial Intelligence and the Internet of Things into processes to enhance efficiency and flexibility. However, analysis of time series sensor data, crucial for process optimization, is susceptible to Data Quality Issues (DQIs) and can lead to operational problems. Traditional Machine Learning approaches struggle with limited error data availability in addressing DQIs. The knowledge-driven approach of Case-Based Reasoning targets this issue by reusing experiences regarding already identified DQIs. While some DQIs can be detected using conventional similarity measures, the common, frequently occurring DQI type of missing sensor values pose challenges that cannot be solved using established measures. To address this, this paper proposes a weight-based extension of similarity measures for time series data. This extension aims at the identification and handling of missing sensor values in smart manufacturing processes. Furthermore, analog extensions of established time series measures are presented and possible areas of application outside the DQI domain are outlined.

Keywords: Temporal Case-Based Reasoning · Time Series Data · Time Series Similarity Measures · Data Quality Issues

1 Introduction

Smart Manufacturing [49] refers to the integration of advanced technologies such as *Artificial Intelligence* (AI) and the *Internet of Things* (IoT) into manufactur-

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-031-63646-2_16

ing processes to increase their efficiency, improve their flexibility and enable data-based decisions. In the course of this development, more companies are integrating sensors into their processes to achieve tighter control and a more profound understanding of the processes, based on which these processes can be optimized [40]. The sensor data generated during the process serves as a valuable resource for analysis and control purposes. However, due to the presence of *Data Quality Issues* (DQIs) [17], the reliability of this data can occasionally be compromised. These can lead to problems during process execution, such as production downtimes due to defects, or falsify process analyses, resulting in possible incorrect action recommendations. Therefore, it is important to identify and eliminate DQIs before an analysis is carried out based on this data. Several types of data quality issues are already investigated in preliminary work [6], such as time shifts, complete missing sensor time series, or missing sensor values in a recorded time series. Traditional approaches focus on *Machine Learning* (ML) methods for this purpose. However, DQIs are a domain in which little error data is available, as they have to be recognized and processed mostly manually [12]. It is often not possible to train suitable ML models on such small databases. Therefore, instead of a data-driven AI method, the usage of a knowledge-driven approach is suitable that does not attempt generalization based on the limited data, but instead directly reuses experience [34]. An established AI method for reusing collected experience is *Case-Based Reasoning* (CBR) [1,4]. To apply this technique, a case base can be created in which similar problem situations with already detected faults are stored. During runtime, queries are generated, and similar cases are retrieved from the case base. If a similar case is found, the solution of the case determines whether the current time series data from the IoT sensors is faulty and, if this applies, how this fault effects resulting event logs or other higher-level systems.

In CBR retrieval of time series, established similarity measures are utilized to assess the resemblance between the time series of a query and that of a case [29]. Some types of data quality issues, such as time shifts and complete missing sensor time series, can be detected and classified using conventional time series measures in combination with traditional local-global approaches [4]. In contrast, individual missing sensor values in a recorded time series pose a particular challenge. These are common across domains and often occur frequently so that it is necessary to identify them. Due to the length of the time series, individual outliers usually do not have a significant impact on the similarity calculation, making it difficult to detect whether this type of DQI is present based on established similarity measures alone. If this data is recorded at fixed intervals, a simple algorithm for determining this type of DQI would be suitable, which would take place outside a CBR system. However, this underlying assumption does not always hold, as in some domains, for example, only value changes are recorded (cf. [29]). While this anomaly of missing values can be easily recognizable for simple value domains (e. g., Boolean values), they are hardly visible for more complex domains such as coordinates. Furthermore, a simple, integrated algorithm would only indicate the presence of such an error without providing information about its possible origin. Identifying such an error using similarity measures in CBR, on the other hand, identifies cases that provide information about the source of the error and thus contain a certain explanation in themselves [45]. This process can also include suggested solutions from past cases that improve the decision-making process and provide insight into possible actions for event log repair [10]. Another advantage of using CBR is its interactive character that enables that humans remain in the loop [51] and stay responsible for the decision, which is based on the results of the CBR application.

Due to the described limitations of traditional similarity measures, it is necessary to investigate suitable similarity measures for the DQI type of missing sensor values. This paper's contribution lies in introducing a weight-based enhancement to current similarity measures for time series, aimed at effectively detecting this specified DQI. These similarity measures are developed based on the requirements of the DQI domain and evaluated based on these. However, the area of application of the extension of the traditional measures is not limited to this use case, but can also be relevant for other domains such as speech recognition, financial data or medical diagnoses.

The further structure of the paper is as follows: In Sect. 2, the theoretical foundations and related work are presented. The use case of the DQI is then presented in Sect. 3. On this basis, the problem of why traditional similarity measures are not suitable is explained, and other possible fields of application that benefit from an extended measure are named. In Sect. 4, the approach for integrating weights into established similarity measures is presented and illustrated using selected measures. Sect. 5 deals with an evaluation of these similarity measures based on DQI data. Finally, a conclusion is drawn in Sect. 6 and an outlook on future work is given.

2 Foundations and Related Work

The application of CBR methods to time series falls within the sub-research area of *Temporal Case-Based Reasoning* (TCBR) [16,23]. This deals with how temporal relationships can be expressed in cases. A case expressing temporal relationships is a sequence of certain attributes related to the time dimension. The most common form of representation for such attributes is a time series [29], which is introduced in Sect. 2.1 and illustrated using an example from the DQI domain. Subsequently, similarity measures are presented and categorized in Sect. 2.2 that are used in CBR to address time series.

2.1 Representation of Time Series Data

For modeling similarity measures, it is necessary to fill the knowledge containers [35] of the vocabulary and, on this basis, of the case base [29]. While there are also rarely used representation forms such as episodes, graphs, and event sequences, time series are an established and most basic representation for a temporal case. A time series stands for a measured real value over a time course,

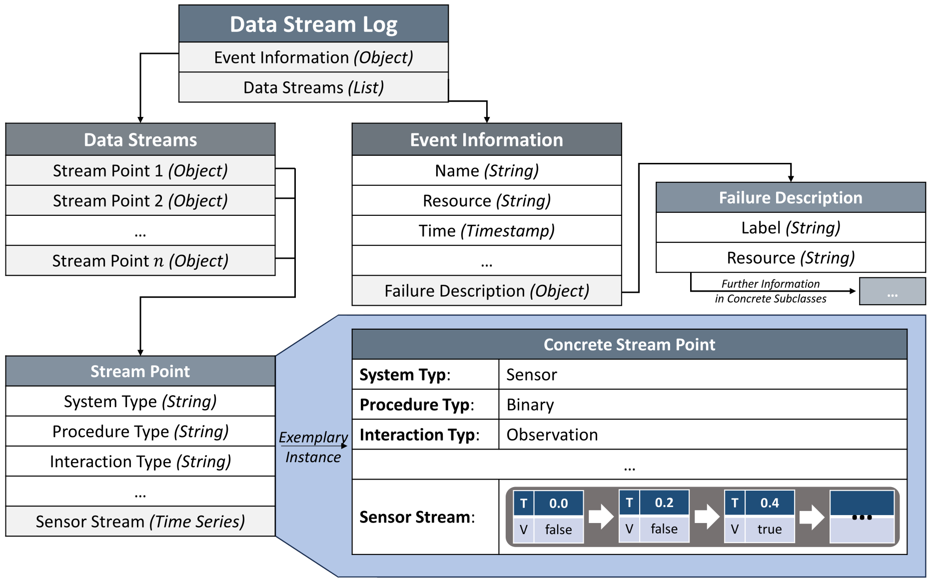

Fig. 1: An Exemplary Object-Oriented Data Stream Case From the DQI Domain Containing Time Series at a Lower Level, with an Exemplary Concrete Instance of a Lower Class.

where concrete time points are referred to. For the IoT domain, present in the DQI use case, such time points are contained in sensor data [12]. To represent the time series, we use a symbolic representation, which can vary depending on the use case, and which represents the real values. These values are summarized and mirrored as a feature vector [29]. There are simplified representation types, such as temporal abstractions [38,43] or Allen intervals [2,16], which are intended to reduce the complexity of the time series and thus that of the subsequent similarity calculation. However, such a change to the time series is not suitable for the DQI use case of missing sensor values, as abstraction results in a loss of information. So, the explicit consideration of the individual time points is no longer possible. Therefore, unchanged feature vectors are used, which can be embedded in other objects at higher levels if necessary. The individual values within the sequence are also objects that contain, on the one hand, the timestamp and, on the other hand, the symbolically represented value at this time.

Fig. 1 shows an example of a vocabulary that originates from the DQI domain. An object-oriented case is shown here, which is an instance of the `Data Stream Log` class. This aggregates several attributes that describe information on the global level of the event (class `Event Information`), which in turn contains a specific instance to describe the DQI (class `Failure Description`). This specific object forms the solution part of the case. The data stream log case also contains several data stream points (class `Stream Point`), which represent a sensor with its attributes and the specific time series. Fig. 1 shows an example

Table 1: The Three Categories of Similarity Measures for Time Series (according to Malburg et al. [29]).

Cat. 1	Cat. 2	Cat. 3
Similarity measures that can only be applied to time series of the same length. These compare only the values at the corresponding times.	Similarity measures that can be applied to time series of different lengths and consider not only the values, but the time points themselves.	Similarity measures, like those in Cat. 2, but that can detect stretching and compression in addition.

instance of such a stream point. This is a sensor that measures the breaking of a light barrier and accordingly contains a time series of Boolean values. This time series consists of attribute-value pairs, each of which contains a time point (here, the time in milliseconds since the start of the process) and the corresponding value. In addition to Boolean values, other time series from other sensors may contain, for example, coordinate values, weight measurements, speeds, or other sensor values. In this specific case representation from the DQI domain, the time series are at the lowest, local level of the vocabulary.

2.2 Similarity Measures for Time Series Data

The calculation of similarity between time series is an area of research that is investigated in CBR [29, 37] as well as in other research areas [13, 22]. In preliminary work [29], we divide the syntactic similarity measures already used in research into three categories that are shown in Tab. 1. For all of these categories, the similarity calculation is based on the local-global principle [4, pp. 106–107]. In this context, similarities are calculated at the level of individual attributes and then aggregated globally for the complete case. Semantic similarity measures can be integrated at the local level to determine the similarities of the individual attribute values and the time points. However, the time series measures do not consider any further semantic information, such as dependencies (cf. [21]) of the time series attributes within cases. In addition, no domain knowledge is included in the measures themselves so that these are knowledge-poor similarity measures [46, pp. 59–62]. In domains such as DQI, in which an anomaly is to be recognized and classified based on semantic information stored elsewhere in the cases, such syntactic similarity measures for time series are sufficient. At the overall case level, the global similarity measure is knowledge-intensively enriched by taking this domain information into account. In preliminary work [29, 37], common similarity measures are identified for each of the categories. These are shown in Tab. 2. For each of these similarity measures, a suitable similarity measure can be used at the local level.

Table 2: Established Algorithms From the Literature [29, 37] for the Three Categories of Similarity Measures for Time Series.

Cat. 1: List Mapping	**Cat. 2**: *Smith-Waterman-Algorithm* (SWA)	**Cat. 3**: *Dynamic Time Warping* (DTW)
To compare two sequences of equal length, a direct mapping of the time series elements is conducted [29], disregarding time points. This method can also assess time series of differing lengths by treating the shorter sequence as a subset of the longer one and finding a corresponding subsequence of equal length.	An algorithm to determine the sequence matching based on required insertion or deletion operators is SWA [37, 44]. This is done based on a scoring matrix in which operations are penalized so that the best possible similarity value can be determined from the matrix.	An algorithm that can also deal with the stretching and compression of time series common in many domains is DTW [36, 37]. Analogous to SWA, scoring matrices are created that determine the steps from one sequence to another. The maximum value is the best possible similarity.

3 Use Case and Requirement For Extended Measures

To understand the limitations of traditional similarity measures and the associated need to extend them, the use case of the DQI domain is first presented in Sect. 3.1. On this basis, the problem of identifying missing sensor values using CBR is introduced in Sect. 3.2 and the necessity for the weight integration is explained. Sect. 3.3 describes further use cases in which the application of weighted time series measures can also be worthwhile.

3.1 Data Quality Issues

Addressing data quality in the IoT domain is a research area that ranges from detecting such data quality issues to addressing them through cleaning methods [6, 17, 48]. In particular, low-cost sensors with limited battery and processing power used in harsh environments can lead to sensor problems [48]. These include failures such as low sensing accuracy, calibration loss, sensor failures, incorrect device placement, range limitation and data packet loss. Such sensor failures in turn cause various types of errors in the generated data, which complicates further analysis. For example, these errors are reflected in the data often as outliers, missing values, bias, drift, noise, constant value, uncertainty or stuck-at-zero. Bose et al. [7] list missing, incorrect, imprecise and irrelevant data as superordinate categories for such DQIs. Verhulst [50] examines other dimensions of DQIs in a taxonomy, such as the completeness or correctness of the time series in event logs. Leaving these errors untreated leads to incorrect data, and the subsequent analysis may provide unreliable results that ultimately could cause incorrect decisions [6]. To avoid wrong decisions, it is important to evaluate the underlying data quality. For this purpose, measures of quality such as completeness, timeliness, plausibility, and concordance are addressed [20].

The DQI failure of missing sensor values occurs if one or more values that should have been observed are not contained in the time series. Therefore, it falls into the category of missing data [7] and addresses a completeness issue [50]. This error can originate from the sensor itself or occur due to a loss in data transmission. Depending on the cause, the failure must be rectified in different ways, for example by manual recalibration or data imputation. The error can become apparent in the final time series in two ways: In the first way, the time series logs the data at fixed intervals and one or more values are missing. In the second way, only value changes are logged, so that missing values can only be noticed if a state transition or value change has not been logged.

3.2 Identification Problem of Missing Sensor Values

Missing values can also occur intentionally due to sensor calibrations or similar, so that it does not necessarily have to be an error that can be determined by a pattern. To identify a suitable, similar error case, similarities to the currently available sensor time series must be calculated. Since a syntactic similarity of the time series is sufficient and semantic information can be taken from the attribute values of the case (see Fig. 1), the similarity measures presented in Sect. 2.2 are suitable in general.

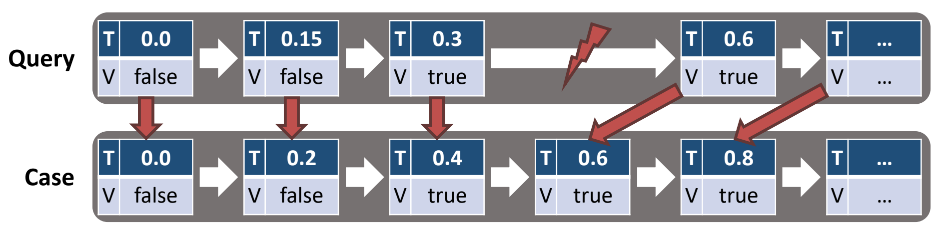

Fig. 2: Mapping of a Time Series With the Missing Sensor Failure to Another, Error-Free Time Series.

Fig. 2 shows two time series, for which a similarity is calculated. A sensor value is missing in the query time series, but not in the one from the case. Here, there would be different mapping methods applied depending on the similarity measure used. With the list mapping from Cat. 1, a similarity comparison of the time series with unequal lengths would not be possible. Due to the different intervals the values are recorded in this example, the application would not be possible even with two error-free time series if both had run for the same length of time. To calculate the similarity of one time series as a subsequence of the other, would not be suited as well, as the different time points would result in an unrealistic similarity value. When using SWA as a Cat. 2 measure, penalties could be applied for insertions or deletions that occur when one time series is to be transferred to the other. Using these, missing values would have a significant impact on global similarity. However, if the penalties were too high, the algorithm would prefer to accept very low local similarities during mapping instead of maximizing the local similarities, as this would nevertheless increase the global similarity. In addition, SWA cannot deal with stretching and compression, which occurs in the example in Fig. 2. DTW as a Cat. 3 measure, on the other hand, would be suitable for mapping the stretched upper time series to the compressed lower time series. Depending on the local similarity measure, the different time points would hardly matter, and the two time series would be identified as similar. However, the missing value would also go unnoticed, as DTW would map several elements of the query to one element of the case and achieve similarity values minimally below 1.0. Even with a selective local similarity measure for the time points and an equal weighting with the actual attribute value there, the resulting relatively low local similarity value would hardly carry any weight in the global time series measure.

As described, none of the time series measures presented is therefore suitable for the use case of the DQI type of missing sensor value. To identify such an error using a time series measure, a further factor is therefore required in addition to a selective local measure enabling a significant influence on the global similarity. The integration of weights into the presented similarity measures is a possibility to achieve the desired selectivity. In the case of missing sensor values, DTW is a suitable similarity measure. However, this is not the only possible use case for this nor the other categories. Since CBR has not yet been used as an anomaly detection method for a use case such as the DQI, there is no time series similarity measure that is suitable for this purpose. As described, this must penalize a small deviation from the good case to such an extent that this deviation manifests itself in the global time series similarity.

3.3 Further Use Cases

In addition to the described missing sensor failure from the DQI domain, possible use cases can be derived for weighted similarity measures of the various categories to detect anomalies. A weighted list mapping as a Cat. 1 measure is suitable for use cases in which time series of the same length are available, but where individual, local differences can be serious. In an industrial context, if two sensors log their time series at the same interval, individual outliers can indicate incorrectly calibrated sensors or other errors. These can be clearly highlighted using weights so that abnormal cases can be better distinguished from error-free cases. Furthermore, in medical monitoring, where time series are already being analyzed using CBR [47], individual outlier values can indicate poor health conditions that would hardly be noticed when using classic measurements. For Cat. 2 with SWA, for example, weights in text syntax checking would make it possible to emphasize errors caused by individual letters or characters when detecting spelling errors. CBR is already used for spell and grammar checking with SWA as one of the measures [3]. In addition, mutations in biology, analogous to the original purpose of SWA [44], could be better distinguished from healthy genes and thus identified. DTW as a measure of Cat. 3 could also be used for the areas of application already covered by the TCBR of speech recognition [26] or motion detection [14], for example, to identify individual errors caused by background noise. Conversely, in the already investigated area of financial data [9, 15], market volatility or seasonal trends, for example, could be intercepted using weights so that these would be less significant in terms of global similarity. Another area in which TCBR is already being used is the home monitoring of elderly people in a smart home [25]. Here, too, short-term anomalies can be highlighted much better using weights. In an industrial environment, this weighted measure may be of interest in the context of predictive maintenance where in the CBR context expert knowledge is already used [19, 29] to detect anomalies and identify their causes based on case knowledge.

The suitability of the measures for the respective domains must be examined and evaluated. This list merely serves as an example to show that this contribution is not limited to the DQI domain and reaches beyond this.

4 Approach of Weighted Time Series Similarity Measures

The similarity measures for time series presented in Sect. 2.2 all have in common that an unweighted mean value is used when aggregating the local similarity values. For longer time series, low local similarity values due to individual deviations have a minimal influence on the global similarity value at the time series level and thus also on possible object levels above it. The SWA approach also prefers to accept poor local similarity values instead of a penalty to maximize similarity, so that higher penalty values have no influence in many domains. Therefore, we introduce weights for the individual local similarities in this contribution, which we present in the following. Thereby, we describe how these weights can be integrated into the respective global time series measures.

Some definitions that are essential for understanding the approach are introduced in the following. The definitions are partly based on the CBR definitions by Bergmann [4, pp. 48–60].

Definition 1 (case) *Let c be a time series case, represented as a tuple (d, l), where d denotes the problem description and l as corresponding solution. For both hold that they are represented by the vocabulary container: $(d, l) \in VOC$. Each element in the time series at a position h is referred to with c_j for which holds $q_j \in c$.*

In the DQI use case, the description contains the measured time series and the meta information, while the solution is the failure classification, such as a missing sensor value (see Fig. 2).

Definition 2 (query) *Let q be the time series query for which a retrieval is executed. Like every case c, q is also represented using the same vocabulary, but contains no solution part, i.e., $q = (d) \in VOC$. Each element in the time series at a position i is referred to with q_i for which holds $q_i \in q$.*

The following definitions are specific to the proposed approach and build on the above definitions. The relationship of an element from the query to a corresponding element of the case is described using a mapping.

Definition 3 (mapping) *Let m be a mapping function $\forall q_i \in q\ \exists c_j \in c\ m(q_i) : q_i \rightarrow c_j$ that maps an element of the query q to a corresponding element from the case c. The function is injective, there is only one mapping partner for each element $q_i \in q$. In turn, several elements from the query can point to $c_j \in c$.*

For time series, such a mapping is always calculated based on the query. Each mapping pair has a local similarity. The aggregation of the local similarities leads to the global similarity value.

Definition 4 (similarity) *Let the similarity between query q and case c be denoted by $sim(q, c) \in [0, 1]$. The function $sim_{local}(q_i, m(q_i)) \in [0, 1]$ is defined as the local similarity of two values for the value $q_i \in q$ with its corresponding mapping partner $m(q_i) \in c$. These local similarities are aggregated in a global time series similarity measure. The value range of each similarity function is bound to the interval $[0, 1]$.*

The local and global similarities are calculated based on functions that assign a similarity value to two elements. At a global level, this is based on the aggregation of the local similarities. For the integration of the weights, the original mappings are calculated by the similarity measures to be extended. For each of the similarity measures presented in Sect. 2.2, the integration is performed in the same way. Therefore, the weights are integrated into the local similarities for the already computed mappings. This is because the mappings would be different if the weights had been applied beforehand, since new local similarities exist. This would contradict the idea of weighted time series measures, that outliers have a greater influence on the global time series similarity due to penalties. Therefore, the mappings and similarities are first calculated based on the traditional measures, only the global aggregation is not carried out.

Definition 5 (traditional similarity measure) *Let $sim_{trad}(q, c)$ be the function for traditional time series similarity measure. For the weighted approach, this similarity measure returns a set containing each mapping with a local similarity value instead of the aggregated global similarity value. Therefore, it holds:*

$$sim_{trad}(q, c) = \{((q_1, m(q_1)), sim_{local}(q_1, m(q_1))), ..., ((q_n, m(q_n)), sim_{local}(q_n, m(q_n)))\}$$

The mapping partners calculated here can be accessed with the function $m(q_i)$. The local similarity values are also accessed with $sim_{local}(q_i, m(q_i))$.

To apply weights depending on local similarity values, weights must be set for intervals of similarity values. These can be defined manually or, depending on the domain, learned by ML methods. Therefore, the following definition is introduced.

Definition 6 (weighted local similarity, maximum weight) *Let $w : [0,1] \to \mathbb{R}_0^+$ be a function. Then w maps from the closed interval 0 to 1 to the set of positive real numbers $\mathbb{R}_0^+$. max_w denotes the maximum weight for a given weight function w, such that for all $x \in [0,1]$, it holds that $w(x) \leq max_w$.*

The sum of the weights does not have to be 1.0 at this point. To implement penalties for low similarity values, high weights are suitable for low similarities and low weights for high similarities. For example, the similarity value interval $[0.0, 0.2]$ can be weighted with a factor of 10, the interval $(0.2, 0.5]$ with a factor of 5, and the interval $(0.5, 1.0]$ with a factor of 1.

Overall, the entire interval range of possible similarity values of $[0.0, 1.0]$ must be covered. A weighting of 0.0 should be assumed for all interval ranges that are not defined. The weights for the individual intervals are normalized using the highest possible weight value so that they are contained in the interval $[0, 1]$. Therefore, the following definition is introduced.

Definition 7 (normalized weights) *The normalized weight for each local similarity value is referred to as $w_{norm}(sim_{local}(q_i, m(q_i)) \in [0,1]$. This is calculated by the following formula:*

$$w_{norm}(sim_{local}(q_i, m(q_i))) = \frac{w(sim_{local}(q_i, c_i))}{max_w}$$

Each normalized weight $w_{norm}(sim_{local}(q_i, c_i))$ is bound after normalization to the interval $[0, 1]$.

For the example, all weights are normalized using the highest weight, in this case 10. The normalized weights are therefore 1.0 for the similarity values in interval range $[0.0, 0.2]$, 0.5 for the range $(0.2, 0.5]$, and 0.1 for the range $(0.5, 1.0]$.

To calculate the global weighted similarity, the local similarity values are recalculated based on the normalized weights, and aggregated to a global similarity value. Therefore, the following similarity function is defined.

Definition 8 (weighted global similarity) *Let the global weighted similarity be referred to as $sim_{weighted}(q, c) \in [0, 1]$. For this, it holds:*

$$sim_{weighted}(q,c) = \frac{\sum\limits_{q_i \in q, m(q_i) \in c} sim_{local}(q_i, m(q_i)) * w_{norm}(sim_{local}(q_i, m(q_i)))}{\sum\limits_{q_i \in q, m(q_i) \in c} w_{norm}(sim_{local}(q_i, m(q_i)))}$$

The application of the formula and the difference it makes is illustrated in the following example. If we have one case with a similarity of 0.1 and five others with 1.0, the traditional average similarity value would be 0.85. The similarity value would be high and would cushion the one outlier. If the weights introduced as examples are used, the similarity value would be reduced. If the weights introduced as examples are used, the similarity value would be reduced. The local similarity for the case with the output similarity 0.1 is still 0.1 as the weight for this interval range is 1.0. For the cases with local similarities 1.0, the local similarity decreases to 0.1 by multiplication with the weight of 1.0 in each case. Thus, the global similarity for the weighted DTW measure is $sim_{weighted}(q, c) = 0.4$. By applying the weights, in this case, to the interval with low similarity values, the penalty ultimately has an impact and allows individual outliers to have a significant influence on the global similarity of two time series.

5 Evaluation

The presented extension of the similarity measures for time series using weights was evaluated using the DQI domain. The hypothesis to be investigated is that the integration of weights enables a better classification of the DQI type of missing sensor values. For this purpose, the underlying data and the methodology of the evaluation are presented in Sect. 5.1. The results of the evaluation are then presented in Sect. 5.2 and discussed in Sect. 5.3.

5.1 Experimental Setup

The presented extensions of the time series similarity measures are implemented in the CBR framework ProCAKE [5], which already contains the traditional measures [42] and is already used for time series applications [24, 29, 37]. The evaluation of these measures takes place in the DQI domain and is therefore limited to the extended version of DTW. The suitability of the other measures must be checked separately. For the evaluation, we used the Fischertechnik Smart Factory from the IoT Lab Trier[1] [30] as application. The error-free data used is publicly available [27] and is represented in the DataStream format [31]. In addition, data was generated containing three different DQI types: Missing sensor value, as well as time shift and missing sensors. A reduced case base was used for the evaluation, which contains 425 error-free cases and 25 cases of each respective DQI type, all randomly chosen. This means that a total of 500 cases are available, 25 of which are cases containing a missing sensor failure. To show the generalizability of this method, a cross validation is performed. A 10-fold is used as a setup which is performed separately for both, the DTW measure and the weighted DTW measure (in the configuration listed as an example in Sect. 4). Based on the five most similar cases and an additional threshold value of 0.9, which ensures that only sufficiently similar cases are considered, a majority vote is carried out to determine whether the failure type is present or not.

5.2 Experiment Results

The evaluation has been carried out on a server with 34 processors, a clock frequency of 2850 MHz, and 400 gigabytes of RAM. As some cases contained

[1] https://iot.uni-trier.de/

several long time series and due to the quadratic runtime of the DTW measure [37], some calculations took a long time. Consequently, a ten-minute time limit per retrieval has been imposed, resulting in the termination of some queries that exceeded this duration. This resulted in 288 retrieval runs of the 500 being carried out. The average time per run has been 8 minutes and 40 seconds. The further 212 cases are therefore not considered in the evaluation.

Based on the most similar cases, a majority vote has been carried out after the retrieval. The resulted classification determined by the CBR system was then compared with the actual presence of the missing sensor failure. The traditional approach has been able to classify two of the 13 DQI error cases correctly and two false-positive misclassifications. The weighted approach, on the other hand, has been able to identify five of the error cases correctly and three false positives under the above conditions. On this basis, accuracy, precision, recall, specificity and F1 score are calculated as performance measures. In terms of accuracy, precision, recall and thus the F1 score, the weighted DTW measure is minimally better, while the specificity for the traditional measure is higher. The comparison shows that while the weighted DTW measure performs better on the correct failure classification measures, it lags the traditional measure on other performance measures. Due to the sample size of 288 requests, of which only 13 are failure cases, the relative numbers are susceptible to small absolute number changes. Overall, the classifications only change in three cases when the different measures are applied.

5.3 Discussion

As explained before, the weighted time series measure improves the classification of missing sensor values compared to the traditional measure. However, it also does not make it possible to find the majority of failure cases. While the traditional DTW measure classifies only two out of 13 failure cases, the weighted measure detects five of them. Thus, eight failure cases are still incorrectly not found, while the risk of false positive classification also increases. Due to these results, which fall short of expectations, the error cases set as queries in the evaluation and their retrieval results were examined more closely. It is noticed that two types can occur:

a) There are one or more error cases that have a missing sensor value at a similar point. The weighted DTW measure can provide discriminatory power by penalizing small deviations from good cases and enable correct classification.
b) There are only fault cases that have missing sensor values in other places. The application of the similarity measure reduces the similarities to all cases, so that the ranking of the most similar cases does not change and at most cases are taken outside by the threshold value.

As the evaluation data set was selected at random, it is not possible to ensure that there were enough cases with similar faults. However, this corresponds to the real conditions in production plants, where faults occur infrequently and have to be recorded manually. Type b) is therefore the one that occurs much more frequently. For this case, the weighted similarity measures are only suitable to a limited extent. It can also happen that the similarity values to actually similar cases also fall due to the high penalties. If it can be ensured that type a) is present, the similarity measure presented is suitable for the application. In general, however, it cannot be shown that the weighted similarity measures are appropriate for the specific use case of missing sensor failures. Therefore, the hypothesis cannot be entirely confirmed, but neither can it be refuted. Further research may need to investigate a hybrid approach [11, 39] combining CBR and ML techniques to investigate how the performance of the measure can be optimized for the DQI domain. At the very least, the weighted DTW measure can be used for DQI to identify a suitable case to explain a failure that has already been identified based on the case [32]. Furthermore, it should also be investigated how the relatively high runtime of the calculations can be reduced to be able to use the procedures for real-time diagnostics. An evaluation of the suitability for other domains, as described in Sect. 3.3, can also be carried out.

6 Conclusion and Future Work

In this paper, the DQI problem of missing sensor values is presented, and it is explained why addressing by CBR is appropriate. Therefore, an approach for the integration of weights into similarity measures for time series is provided. For each category of similarity measures from the TCBR domain, the integration is presented conceptually as well as prototypically implemented and evaluated in ProCAKE. For the DQI use case, it is shown that these adapted similarity measures enable a more accurate identification of anomalies due to missing values, as well as classification. However, this improvement is not sufficient to reliably detect such failures in time series, if there are not enough suitable error cases. As this is not the case in most use cases, further optimizations of this approach or research into other methods are necessary.

From the evaluation of this measure, the runtime optimization of the time series measures is derived as further research potential. For other case representations, GPU methods are used for retrieval in preliminary work [28], which significantly accelerated the retrieval phase. It can be investigated how these can be adapted to time series and what influence they have on the retrieval for this case representation. Distributed CBR approaches [33] can also be used to shorten the computing time. Such distributed computing in edge-cloud architectures is particularly suitable for production environments [40]. Alternatively, case-based maintenance methods [8] can also be used for runtime optimization, which can be investigated for the DQI domain or TCBR in general. Another possibility would be to research a hybrid CBR approach [39]. This could integrate embeddings as ML methods that are trained based on the weighted DTW measure, thereby significantly accelerating retrieval. Similar approaches are already applied to complex similarity measures for other case representations [18].

Additional to this, we want to extend the DQI investigation by designing a CBR framework that provides to analyze sensor data for different failure types, such as additional time shifts and complete missing sensor time series. This CBR construct is expected to further elaborate the already presented advantages of CBR over ML methods due to the small amount of data. Within this framework, the similarity measures presented in this paper or an extended version may be applied to identify the one type. In addition, an explanatory component can be investigated, analogous to approaches for other representations [41]. Furthermore, it can be researched whether the extended weighted similarity measures presented in this work can also be used for other types of anomaly detection, as presented in this contribution. Hybrid AI approaches are also named as possible extensions to be investigated. As an alternative to the presented weighted measure, embeddings for the detection of missing sensor values can be investigated by training them to recognize this type of DQI and then be included as a similarity measure in CBR.

Acknowledgments. This work is funded by the Federal Ministry for Economic Affairs and Climate Action under grant No. 01MD22002C *EASY*.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994)
2. Allen, J.F.: Maintaining Knowledge about Temporal Intervals. Commun. ACM **26**(11), 832–843 (1983)
3. Bacca, J., Baldiris, S., Fabregat, R., Ávila, C.: A Case-based Reasoning Approach to Validate Grammatical Gender and Number Agreement in Spanish language. Int. J. Interact. Multim. Artif. Intell. **2**(1), 73–81 (2013)
4. Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications, LNCS, vol. 2432. Springer (2003)
5. Bergmann, R., Grumbach, L., Malburg, L., Zeyen, C.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: 27th ICCBR Workshop Proc. vol. 2567, pp. 156–161. CEUR-WS.org (2019)
6. Bertrand, Y., Belle, R.V., Weerdt, J.D., Serral, E.: Defining Data Quality Issues in Process Mining with IoT Data. In: ICPM 2022 Workshops. LNBIP, vol. 468, pp. 422–434. Springer (2022)
7. Bose, J.C.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna improve process mining results? In: CIDM 2013 Proc. pp. 127–134. IEEE (2013)
8. Chebel-Morello, B., Haouchine, M.K., Zerhouni, N.: Case-based maintenance: Structuring and incrementing the case base. Knowl. Based Syst. **88**, 165–183 (2015)
9. Chun, S.H., Jang, J.W.: A New Trend Pattern-Matching Method of Interactive Case-Based Reasoning for Stock Price Predictions. Sust. **14**(3), 1366 (2022)
10. Conforti, R., La Rosa, M., ter Hofstede, A.: Timestamp Repair for Business Process Event Logs. Tech. rep., University of Melbourne: Melbourne, Australia (2018)
11. Corchado, J.M., Lees, B.: Adaptation of Cases for Case Based Forecasting with Neural Network Support. In: Soft Computing in CBR, pp. 293–319. Springer (2001)
12. Grüger, J., Malburg, L., Bergmann, R.: IoT-enriched event log generation and quality analytics: a case study. it - Information Technology (2023)

13. Gunopulos, D., Das, G.: Time Series Similarity Measures and Time Series Indexing. In: SIGMOD'01 Proc. p. 624. ACM (2001)
14. Ho, Y., Sekine, N., Sato-Shimokawara, E., Yamaguchi, T.: Motion pattern recognition using case-based reasoning for information providing. In: SICE Annual Conference 2011. pp. 1284–1289. IEEE (2011)
15. Ince, H.: Short term stock selection with case-based reasoning technique. Appl. Soft Comput. **22**, 205–212 (2014)
16. Jære, M.D., Aamodt, A., Skalle, P.: Representing Temporal Knowledge for Case-Based Prediction. In: 6th ECCBR Proc. LNCS, vol. 2416, pp. 174–188. Springer (2002)
17. Karkouch, A., Mousannif, H., Al Moatassime, H., Noel, T.: Data Quality in Internet of Things: A state-of-the-art survey. JNCA **73**, 57–81 (2016)
18. Klein, P., Malburg, L., Bergmann, R.: Learning Workflow Embeddings to Improve the Performance of Similarity-Based Retrieval for Process-Oriented Case-Based Reasoning. In: 27th ICCBR Proc. LNCS, vol. 11680, pp. 188–203. Springer (2019)
19. Klein, P., Weingarz, N., Bergmann, R.: Enhancing Siamese Neural Networks Through Expert Knowledge for Predictive Maintenance. CCIS, vol. 1325, pp. 77–92. Springer (2020)
20. Kuemper, D., Iggena, T., Toenjes, R., Pulvermüller, E.: Valid.IoT: a framework for sensor data quality analysis and interpolation. In: 9th MMSys Proc. pp. 294–303. ACM (2018)
21. Kumar, R., Schultheis, A., Malburg, L., Hoffmann, M., Bergmann, R.: Considering Inter-Case Dependencies During Similarity-Based Retrieval in Process-Oriented Case-Based Reasoning. In: 35th FLAIRS Proc. FloridaOJ (2022)
22. Lhermitte, S., Verbesselt, J., Verstraeten, W.W., Coppin, P.: A comparison of time series similarity measures for classification and change detection of ecosystem dynamics. RSW **115**(12), 3129–3152 (2011)
23. López, B.: Case-Based Reasoning: A Concise Introduction. Synth. Lect. Artif. Intell. Mach. Learn., Morgan & Claypool Publishers (2013)
24. López, B., Malburg, L., Nkisi-Orji, I., Palihawadana, C., Reuss, P., Schönborn, J., Schultheis, A.: Tutorial on CBR Frameworks. In: 31st ICCBR Workshop Proc. vol. 3438, pp. 216–217 (2023)
25. Lupiani, E., Juarez, J.M., Palma, J., Marin, R.: Monitoring Elderly People at Home with Temporal Case-Based Reasoning. KBS **134**, 116–134 (2017)
26. Maier, V., Moore, R.K.: The case for case-based automatic speech recognition. In: 10th INTERSPEECH Proc. pp. 3027–3030. ISCA (2009)
27. Malburg, L., Grüger, J., Bergmann, R.: An IoT-Enriched Event Log for Process Mining in Smart Factories (Zenodo, 2023). https://doi.org/10.6084/m9.figshare.20130794
28. Malburg, L., Hoffmann, M., Trumm, S., Bergmann, R.: Improving Similarity-Based Retrieval Efficiency by Using Graphic Processing Units in Case-Based Reasoning. In: 34th FLAIRS Proc. FloridaOJ (2021)
29. Malburg, L., Schultheis, A., Bergmann, R.: Modeling and Using Complex IoT Time Series Data in Case-Based Reasoning: From Application Scenarios to Implementations. In: 31st ICCBR Workshop Proc. vol. 3438, pp. 81–96 (2023)
30. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using Physical Factory Simulation Models for Business Process Management Research. In: BPM 2020 Workshop Proc. LNBIP, vol. 397, pp. 95–107. Springer (2020)

31. Mangler, J., Grüger, J., Malburg, L., Ehrendorfer, M., Bertrand, Y., Benzin, J.V., Rinderle-Ma, S., Serral Asensio, E., Bergmann, R.: DataStream XES Extension: Embedding IoT Sensor Data into Extensible Event Stream Logs. Future Internet **15**(3) (2023)
32. Olsson, T., Gillblad, D., Funk, P., Xiong, N.: Case-Based Reasoning for Explaining Probabilistic Machine Learning. IJCSIT pp. 87–101 (2014)
33. Plaza, E., McGinty, L.: Distributed case-based reasoning. Knowl. Eng. Rev. **20**(3), 261–265 (2005)
34. Power, D.J.: Decision Support Systems: Concepts and Resources for Managers. Quorum Books (2002)
35. Richter, M.M.: Knowledge Containers. In: Readings in Case-Based Reasoning. Morgan Kaufmann Publishers (2003)
36. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Trans. Aco. Sp. and Sig. Proc. **26**(1), 43–49 (1978)
37. Schake, E., Grumbach, L., Bergmann, R.: A Time-Series Similarity Measure for Case-Based Deviation Management to Support Flexible Workflow Execution. In: 28th ICCBR Proc. LNCS, vol. 12311, pp. 33–48. Springer (2020)
38. Schmidt, R., Gierl, L.: A prognostic model for temporal courses that combines temporal abstraction and case-based reasoning. Int. J. Medical Informatics **74**(2-4), 307–315 (2005)
39. Schultheis, A.: Exploring a Hybrid Case-Based Reasoning Approach for Time Series Adaptation in Predictive Maintenance. In: 32nd ICCBR Workshop Proc. vol. 3708, pp. 230–235. CEUR-WS.org (2024)
40. Schultheis, A., Alt, B., Bast, S., Guldner, A., Jilg, D., Katic, D., Mundorf, J., Schlagenhauf, T., Weber, S., Bergmann, R., Bergweiler, S., Creutz, L., Dartmann, G., Malburg, L., Naumann, S., Rezapour, M., Ruskowski, M.: EASY: Energy-Efficient Analysis and Control Processes in the Dynamic Edge-Cloud Continuum for Industrial Manufacturing. Künstliche Intelligenz (2024), Accepted for Publication.
41. Schultheis, A., Hoffmann, M., Malburg, L., Bergmann, R.: Explanation of Similarities in Process-Oriented Case-Based Reasoning by Visualization. In: 31st ICCBR Proc. LNCS, vol. 14141, pp. 53–68. Springer (2023)
42. Schultheis, A., Zeyen, C., Bergmann, R.: An Overview and Comparison of Case-Based Reasoning Frameworks. In: 31st ICCBR Proc. LNCS, vol. 14141, pp. 327–343. Springer (2023)
43. Shahar, Y.: A Framework for Knowledge-Based Temporal Abstraction. Artif. Intell. **90**(1-2), 79–133 (1997)
44. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. JMB **147**(1), 195–197 (1981)
45. Sørmo, F., Cassens, J., Aamodt, A.: Explanation in Case-Based Reasoning – Perspectives and Goals. Artif. Intell. Rev. **24**(2), 109–143 (2005)
46. Stahl, A.: Learning of knowledge-intensive similarity measures in case-based reasoning. Ph.D. thesis, University of Kaiserslautern (2004)
47. Szczepanski, T., Bach, K., Aamodt, A.: Challenges for the Similarity-Based Comparison of Human Physical Activities Using Time Series Data. In: 31st ICCBR Workshop Proc. vol. 1815, pp. 173–177 (2016)
48. Teh, H.Y., Kempa-Liehr, A.W., Wang, K.I.: Sensor data quality: a systematic review. J. Big Data **7**(1), 11 (2020)
49. Tran, K.P.: Artificial Intelligence for Smart Manufacturing: Methods and Applications. Sensors **21**(16), 5584 (2021)
50. Verhulst, R.: Evaluating Quality of Event Data within Event Logs: An Extensible Framework. Master's thesis, Eindhoven Univ. of Technology, Netherlands (2016)

51. Zanzotto, F.M.: Viewpoint: Human-in-the-loop Artificial Intelligence. J. Artif. Intell. Res. **64**, 243–252 (2019)