Big Data Research 26 (2021) 100251

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Visual Exploration of Anomalies in Cyclic Time Series Data with Matrix and Glyph Representations

Josef Suschnigg [a,b,*], Belgin Mutlu [a,c], Georgios Koutroulis [a], Vedran Sabol [c], Stefan Thalmann [d], Tobias Schreck [e]

[a] Pro2Future GmbH, Inffeldgasse 25F, 8010 Graz, Austria
[b] AVL List GmbH, Graz, Austria
[c] Know-Center GmbH, Graz, Austria
[d] University of Graz, Austria
[e] Graz University of Technology, Austria

A R T I C L E I N F O

Article history:
Received 1 October 2020
Received in revised form 15 March 2021
Accepted 20 June 2021
Available online 16 August 2021

Keywords:
Anomaly detection
Sensor data
Visual analytics
Visual exploration

A B S T R A C T

The digitalization of manufacturing involves machines equipped with sensors that collect, produce, and exchange data machine-to-machine and machine-to-human in real-time. As the data generated within a production process can be massive and overwhelming for human users, support is needed to understand and explore this data, and drive decisions from it. First, the data has to be monitored and recorded using methods that can handle massive datasets. Next, the collected data has to be analyzed (often in real-time) to, e.g., (i) identify undetected process correlations, (ii) forecast the product quality, and (iii) perform root-cause analysis of failures or problems. The analysis becomes even more valuable when the production process is divided into repeating tasks, producing a vast amount of comparable data. For instance, in automotive durability tests, engineers investigate an engine's condition using multiple sensors, recording data from repeating test cycles. Tests can span dozens or hundreds of cycles, and thousands of runtime hours, making it difficult for engineers to collect and monitor each iteration's data to detect interesting data, such as anomalies. We propose an interactive visual analytics approach that displays the iterations of durability tests as a collection of color-encoded cycle glyphs to tackle this issue. With our approach, domain users including test engineers can readily monitor the test, detect potential anomalies, and intuitively analyze, report and document the detected anomalies. This research is conducted in close collaboration with our partner from the automotive sector and shows the effectiveness and efficiency of a prototype with a pair analytics evaluation study. We open up directions for future work, including a visual interactive labeling concept for anomaly classification.

1. Introduction

This work is an extension of work originally presented by Suschnigg et al. [1] at the BigVis 2020 Workshop held in conjunction with the 23rd Intl. Conference on Extending Database Technology (EDBT 2020) & 23rd Intl. Conference on Database Theory (ICDT 2020). To distinguish this research paper from the original work we state the extensions as follows: (1) a substantially revised title, abstract and introduction, (2) an additional anomaly detector, (3) a more detailed evaluation section, (4) a more detailed discussion section with additional results, (5) a more extensive outlook for future work and (6) concept and preliminary results on the ongoing work on visual interactive labeling for anomaly classification.

Digitization changes the way how manufacturing is operated. Machines equipped with growing amounts of interconnected sensors can capture more and more details of production and other industrial processes [2]. Further, data from industrial processes are increasingly exchanged within supply-chains leading to digital supply chains [3]. As a result, massive amounts of data are available, and the ability to use this data becomes a central success factor for manufacturing companies [4]. Due to the complexity of the data sets and the interwoven application contexts, manufacturing experts need cognitive decision support [5]. However, for human perception, it can be overwhelming to observe and analyze large industrial data sets. Another essential requirement of analyzing industrial data is that extensive professional and domain-specific knowledge of users is required. Visual analytics research

* Corresponding author at: Pro2Future GmbH, Inffeldgasse 25F, 8010 Graz, Austria.
E-mail address: josef.suschnigg@pro2future.at (J. Suschnigg).

https://doi.org/10.1016/j.bdr.2021.100251
2214-5796/

proposes tools supporting domain experts to explore large and complex data sets to address those challenges [6]. Big data research states that data science processes are highly iterative and exploratory, whereas, for many industrial applications, there is no single model or algorithm that can handle all data set varieties and changes in data that may occur over time [7]. An ongoing research challenge in big data visualization highlights that analytics require a combination of visual, interactive, and automated analysis methods [8]. Using this diversity of methods to provide an effective and efficient tool, we propose a glyph-based visual analytics approach to explore and analyze anomalies in multivariate cyclic industrial time series data. The idea is based on a design study, conducted in strong collaboration with an industry partner from the automotive sector, focusing on powertrain testbeds. The main driver of the focus on anomaly detection is the common occurrence of real-world problems in automotive condition monitoring.

This paper's contributions include: (1) Characterization of an industrial data set and a design study. (2) An extendable and versatile glyph-based visual analytics approach for anomaly detection in multivariate sensor data. The application of techniques enabling users to visually identify conspicuous sensor data by a matrix representation for drill-down and further comparative analysis. (3) Results and discussion of a pair analytics evaluation [9], which has been conducted in collaboration with the target user group on the given use case data set. (4) Promising directions for future work, including preliminary results on a visual interactive labeling concept for anomaly classification.

2. Related work

This section discusses the related work conducted in analyzing data using either automated data analysis or visual analytics approaches. Furthermore, we detail the glyph representation as this technique is proven to be an effective manner to represent time series.

2.1. Automated data analysis approaches for anomaly detection

One property of many typical industrial applications is the repetition of specific tasks. To give an example, Maier et al. [10] emphasized reoccurring processes (cycles) for automation and production. In theory, data generated within such cycles should be highly comparable for anomaly detection. Anomalies are generally understood to represent patterns in data that do not conform to a well-defined normal behavior [11]. The literature provides a comprehensive collection of algorithms for the detection of anomalies in multivariate time series data. Anomaly detection also often refers to the term novelty detection [12] or semi-supervised learning [13], whereas for those methods, the definition of normal or rather reference data is needed. Anomaly detection in time series [14] found attention from the industry for several applications, such as predictive maintenance [15], condition monitoring [16] or decision support systems [17]. This work uses three groups of anomaly detection algorithms. The first group covers correlation-based approaches, which are applied effectively on industrial sensor data [16]. This approach's idea is that changes in the bivariate correlation between two sensors can be interpreted as an anomaly. The second group are regression-based methods [18] or reconstruction-based novelty detection methods [12]. This type of anomaly detection's basic idea is to build a regression model on reference (normal) data. The model's estimation errors compared to measured data are rated as anomalies if they exceed a predefined threshold. The third group derives from feature-based classification [19]. They focus on finding a compact description of time series through features and detect outliers in the created feature space [20].

2.2. Visual analytics for industrial application

A survey on visualization and visual analytics applications for smart manufacturing has been published by Zhou et al. [21]. It reveals the diversity of several studies performed for industrial applications and the need for visual analytics. A few examples are available on how to find anomalies in multivariate time series data by visual analytics. An application for finding anomalies in the power consumption of buildings has been proposed by Janetzko et al. [22]. It suggests a model-based and a similarity-based anomaly score and visualizes them in several visualization techniques such as recursive patterns [23], spiral graphs [24], and line charts. In the work of Wu et al. [25], anomalies are detected for condition monitoring by a model-based approach. The deviation of estimated and real values is visualized in a river plot view [26]. As an ongoing challenge, the authors outline the analysts' problem to trust and use the algorithms for condition monitoring. Many different algorithms are available for several applications, and finding appropriate models and parameters is a challenging task. Xia et al. [27] proposed a visual analytics application to support users in finding the right model for dimensionality reduction. Another work addressing this challenge is presented by the EnsembleLens [28]. It is a visual analytics system to help data mining experts to evaluate, compare, and select available anomaly detection algorithms.

2.3. Glyph representation of cyclic time series data

Besides the significant summaries and surveys [29] [30], recently, a systematic review of experimental studies on data glyphs has been presented by Fuchs et al. [31]. Glyphs are an appropriate choice and can enable a quick visual comparison of data values over time [32] to visualize multivariate time series data. Ward and Lipchak [33] proposed the visualization of a circular glyph for recognition of the evolution of measurement of interest. Another glyph-based design for outlier detection in social networks has been proposed by Cao et al. [34]. In their work, glyphs visualize users' suspicious behavior, based on the z-score of several attributes, in a design similar to star glyphs. The anomaly scores of entities are visualized by the red color's intensity in the cyclic glyph center. As examples for glyph-based time series visualizations, a few techniques for glyph designs for comparison purposes are evaluated in the work of Fuchs et al. [32].

3. Background on automotive testbeds and use case

Our use case focused on automotive engines in the context of the validation and verification phase of the industrial product life cycle [35] [36]. After an engine has been developed, its requirements are verified and validated in automotive testbed environments. These requirements can be functional, such as the engine power density, speed, and durability, or of legal nature, such as, along with others, fuel economy, noise pollution, and exhaust gas emissions. For our research, we analyzed data from a durability test of an internal combustion engine. The test's primary goal is to ensure the engine's durability, reliability, and lifetime expectations. Therefore, durability tests are conducted to let the engine undergo sufficiently high mechanical and thermal loads (stresses) and a sufficient number of fatigue cycles (e.g., hundreds of hours) [37]. During durability tests, a vast amount of data is collected by sensors, which are commonly integrated in modern vehicles and accessed through the engine control unit (ECU), or sensors that have been mounted on the engine and the testbed for testing purposes.

Throughout durability tests, engineers observe the test and are responsible for the testee's performance and condition. For the

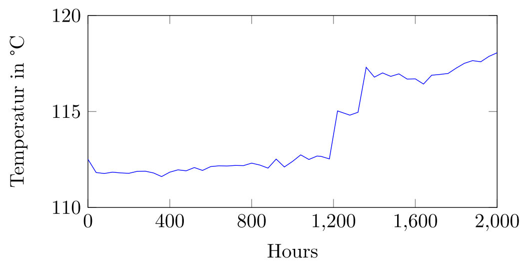

Fig. 1. Temperature of the critical component over all cycles. For each cycle over the whole durability test time the mean temperature has been calculated for temporal trend analysis.

condition monitoring task, engineers generally monitor a few familiar sensors for threshold violations, manually selected and defined by their given domain knowledge or by the customer. However, for novel engine design, which has been recently developed, there is no knowledge on all sensors and their thresholds, and earlier experiences are often not applicable. An essential task in the testing of novel engine designs is a derivation analysis with previous designs. Therefore, our work is motivated by using the time series data of all sensors and the information they might contain.

To give a practical example of the challenges engineers face during durability tests, in Fig. 1 a line plot describing the problem of a use case is shown. After 1,200 hours of 2,000 hours durability test, a fatal error occurred, which increased the temperature of a critical part of the engine by up to 8 °C over time. Finally, the durability test failed because of this temperature increase of a critical component. The failure could not be anticipated for two reasons: (1) the anomaly only occurred in a specific context (at a specific engine speed) and (2) the number of sensors was too large to monitor the whole data set accurately. Consequently, it was challenging to define rule-based thresholds or measurements of concern, indicating the failure upfront. However, compared to a simple rule-based anomaly detection approach, more complex data analysis and models that also take the interplay of sensor data into account may lead to better analysis results. Domain experts assume that it should be possible to anticipate such failures by advanced data analysis and visualization techniques. Therefore, to understand engineers' current data analysis workflow, we carried out a design study forming the basis for our proposed visual analytics application. More details on the data and the design study will be given in the next section.

4. Design study

As the first contribution of this paper, we characterize testbed data and studied how engineers fulfill their condition monitoring tasks through data analysis. This section relates to Miksch and Aigner's "design triangle" [38] and is generally based on the design study methodology of Sedlmair et al. [39]. For the aspects of the tasks of the design triangle, we bridge from goals to tasks with the design study analysis report as proposed by Lam et al. [40].

4.1. Data

Input data used in our proposed visual analytics approach for anomaly detection is taken from an automotive engine testbed. One common task within engine development is to carry out durability tests. For such a durability test, a test cycle is specified to verify the durability of an engine. The test cycle, which is defined by given engine speed and engine torque profile over time, is repeated in a period of several months until the target operating hours are reached. During a durability test, hundreds of sensor measurement signals are acquired and stored continuously, while the engine drives the given profile. In the automotive domain, those sensor measured time series are called channels, which we adopted throughout our research. Among others, channels mainly record several engine speed, engine torque, temperatures, pressures, and exhaust gas measures.

One cycle is stored within one file and can be seen as a $N \times n$-dimensional matrix, where N is the number of channels and n the length of the time series. All signals originally are recording numerical values in a frequency of $10Hz$. Note that channels are aligned according to the given engine speed profile, and therefore, cycles of the same length can be extracted. The target data contains records of $c = 860$ cycles of $2,000$ hours' durability test, in which each cycle has a duration of 140 minutes. Overall, the dataset has 860 cycles x 480 channels x $84,000$ numerical values.

4.2. Users

Users of our proposed visual analytics application are development engineers with a mechanical engineering background, working with powertrains and engines regularly. They have longstanding experience with engines in testbed environments and as front-line analysts, also practically analyzing data to achieve their analysis goals (i.e., condition monitoring). Three users collaborated in our project systematically by participating in the design study and the pair analytics evaluation (see Section 4 and Section 7). In general, testbed data is essential for development engineers and offers the opportunity to measure indicators regarding functional or legal requirements and engine performance. During our research work, we collaborated with data scientists who are daily working with testbeds and powertrains. They continually provided informal feedback from a different view throughout our work.

4.3. Tasks

This design study is based on the domain question if an engine is non-critical during a durability test. For that purpose, we define test cycles as the population unit (or entity, or unit of analysis) [40]. Furthermore, engineers have a high understanding of using cycles as a granularity level for their analysis. Due to cycles' repetitive behavior, they are highly comparable, and therefore we define engineers' data analysis goals as multiple population analysis. Consequently, we identified that engineers were pursuing all three multiple population goals defined in the design study analysis report framework [40] and summarized them in Table 1: (a) compare entities (b) explain differences, and (c) evaluate hypothesis. Columns contain analysis goals and the associated input data. Analysis steps and outputs are in the rows. Overall, data analysis is done by visualizing and comparing time series line plots of sensors, combined with domain knowledge to interpret findings. In the following, we further investigate these goals' characterization by their input, output, and analysis steps.

4.3.1. Compare entities

Engineers attempt to detect population differences as their top-level analysis goal. To achieve that, engineers are observing trends of several familiar channels by calculating the mean values of channels per cycle and visually exploring changes over cycles in a time-ordered line plot (see Fig. 1). In temporal trend charts, up to ten time series are compared either in juxta- or superpositioned line plots [41]. The output of the compare entities analysis goal is to observe a conspicuous temporal trend or anomaly, which is investigated in the explain differences goal.

4.3.2. Explain differences

Regarding the domain question, the output of the explain differences goal can be either that the observation is not relevant for

Table 1
Design study analysis report: Goals and tasks for automotive testbed condition monitoring.

	(a) Compare entities	(b) Explain differences	(c) Evaluate hypothesis
Input	Testbed cycles, whereas within a daily iteration the focus is to compare current cycles with previous cycles	Whole time series and an observation (i.e., anomalous sensor data)	Hypothesis: A conspicuous component caused the anomaly
Steps	Comparing line plots of well known sensors. Up to ten lines and their according scales are visualized in many plots at once. Also, temporal trend analysis of aggregated multiple cycles are visualized by line plots.	Domain knowledge of testbed engineers enables the exploration of specific sensor data line plots to retrace occurrences of observations	Domain knowledge of users allows the investigation of the hypothesis through the data and specific channel line plots
Output	Observation of anomalies and deviations of current daily data regarding past data	Identification of the component on the engine, which is responsible for the anomaly or time frame, when the anomaly happened or started	As a confirmation or rejection of the hypothesis, the conspicuous component is in a bad condition, or not

the engine condition, or as a hypothesis one specific component of the engine is in bad condition. In the next analysis goal, the hypothesis needs to be evaluated.

4.3.3. *Evaluate hypothesis*

Regarding the domain question, the output of the explain differences goal can be either that the observation is not relevant for the engine condition, or as a hypothesis one specific component of the engine is in bad condition. In the next analysis goal, the hypothesis needs to be evaluated.

Through that characterization of higher-level analysis goals, we can derive lower-level task definitions **T1–T5** to address them in our visual analytics design considerations and automated data analysis:

T1 **Identify population contrasts.** Test cycles are the unit of analysis, or population, for engineers. As the first task, population contrasts or differences are explored. This is achieved by temporal trend analysis of a few familiar channels. The problem of dealing with big channel amounts engineers face should be considered in the design by taking all channels into account.

T2 **Application and visualization of semi-supervised anomaly detection methods.** Engineers detect interesting patterns and anomalies mainly by visually exploring line plots of channels. Comparing past cycles to current cycles is related to a semi-supervised learning scenario and should be considered for the choice of automated data analysis and visualization. Automated data analysis should be applied to highlight interesting or conspicuous channels in the visualization. We also assume that the combination of several anomaly detection algorithms leads to more significant findings, for which reason an ensemble method [28] should be considered for the visualization.

T3 **Examine conspicuous channels in multiple populations.** After conspicuous channels have been detected by temporal trend analysis, engineers drill-down to examine and find differences between channel line plots of different populations. The comparison of interesting channels in different populations should be considered in the design.

T4 **Detection of conspicuous channel relations.** With exceptions on visualizing multiple trend line plots in juxtaposition, engineers generally detect anomalies by univariate time series analysis of multiple channels. They also compare line plots with the given engine speed and engine torque. Considering relations between channels at a broader scale should be considered for automated data analysis and visualization.

T5 **Reduce amount of data.** Data reduction techniques should be taken into account for the visual analytics design to address many channels. The primary consideration for the visual analytics design is that interesting data should be highlighted to support engineers in their decision making.

Overall, data can only be analyzed by including extensive domain knowledge of users to the data analysis. Modern powertrains and engines are highly complex machines, and therefore domain experts are necessary to interpret results of automated data analysis through a visual analytics approach. In the next section, we introduce the anomaly detection methods we applied to the visual analytics approach.

5. Automated data analysis: anomaly detection methods

In research, anomaly detection often refers to a two-class classification problem, in which data either is classified as an anomaly or not. In general, a model is built on normal data, considering that the model can calculate an anomaly score on unseen data sets (apart from unsupervised methods). If the anomaly score exceeds a predefined threshold, the data record or the entire set is classified as an anomaly. We consider the application of semi-supervised anomaly detection methods to our design **T2**. Most techniques are specific to different observational features, in consequence of which we assume that an ensemble-based approach obtains more robust anomaly scores [28]. Therefore, we propose to map the results of different anomaly detection algorithms to a unified value for comparison purposes and describe three anomaly detection methods used in the visual analytics approach.

5.1. *Unified anomaly score*

Test cycles are the engineer's unit of analysis, for which reasons we choose them as the granularity level for data analysis (**T1**). To make different anomaly detection methods comparable in an ensemble-based approach, we propose the following to map anomaly scores to unified values between 0 and 1: (1) Interactively select a reference cycle as input data for the training of the anomaly detection model. (2) A baseline cycle is selected to calculate a baseline anomaly score. (3) Define a threshold anomaly score based on prior knowledge, domain knowledge, or historical data. (4) Further, the cycles' anomaly score is calculated as the linear scaling from 0 (baseline) to 1 (threshold). Therefore, our approach needs the definition of a reference cycle for model training, a cycle for baseline definition, and the definition of a threshold. The baseline anomaly score is used to consider a training error and, therefore, is taken as the lower limit of the unified anomaly score calculation. Note that the baseline anomaly score is calculated by a baseline cycle, in which data should be recorded temporally close to the reference cycle. Hence, we specify the cycle subsequent to the reference as the baseline cycle. Defining the upper limit (threshold) is critical and can be changed interactively in the visualization. In the following, we discuss three methods, which have been applied to industrial sensor data in prior research. Another criterion for selecting these methods is the capability of identifying conspicuous channels separately for further drill-down and comparative analysis (**T3**).

5.2. Correlation-based anomaly score

Inspired by the approach presented by Zhao et al. [16], we assume that the change of linear correlations between two sensors refers to an anomaly. During our research work, we investigated the application of correlation-based anomaly detection on testbed data. Despite its limitations, the method also has strengths that help solve the tasks defined in the design study. First, we highlight the main limitation in detecting anomalies by the change of linear sensor correlations, if throughout the durability test, no linear correlation between two specific sensors exists. Nevertheless, we examined that in testbed data, many linear correlations between sensors exist. For example, data from several temperature channels are likely to correlate. Experiments demonstrated that this method could detect anomalies in testbed data, and therefore has been applied to our visual analytics approach.

The basis for the correlation-based anomaly score is the correlation difference matrix, which represents the deviation of linear channel relations between two testbed cycles. The correlation matrix for each sensor combination of the reference cycle, the baseline cycle, and the unseen cycle is calculated by using Pearson's correlation coefficient. Then, the correlation matrix of the unseen cycle is subtracted by the correlation matrix of the reference cycle, which results in the correlation difference matrix. As a result, the anomaly score is calculated as the average of all values in the difference matrix and is mapped to the unified anomaly score accordingly to the method explained before.

5.3. Regression-based anomaly score

As the second anomaly score, we make use of regression models for regression-based anomaly detection [18]. For this research work, we train regression models to estimate a time series. Consequently, the model is applied to an unseen data set, in which the difference between the estimation and the real values (residuals) can be interpreted as anomalies. Considering this method for **T1** and **T2**, an anomaly score between populations or cycles needs to be calculated in a semi-supervised manner. Therefore, regression models with data of a user-defined reference cycle are trained for all channels separately. It is necessary first to standardize data, i.e., standardization of the entire time series to values between 0 and 1, to make those channel regression models comparable. The regression models can now be used to estimate all channel time series for unseen cycles, and the anomaly score of one cycle can be calculated by the root mean squared error over all channels of a cycle. In the following, it can be mapped to a unified anomaly score accordingly to the method explained above.

We chose Random Forest regressor, as suggested by Breiman [42], considering that this model has been proven to perform well in many domains [43]. As input data for training channel regression models, engine speed and engine torque are chosen, since these two channels are given by the test and strongly relate to the majority of the channels (**T4**). Also, sliding window features for these two channels are extracted, whereas sliding windows contain differences and mean values of three seconds into the past. We assume that in this time frame, the most relevant information can be extracted for our models. This approach aims not to estimate each channel as accurate as possible, but to detect a change of anomaly scores between populations. As the correlation-based method, we are aware of the limitation that this method may not return a decent estimation for all channels, but it may be effective for some types of anomalies. This consideration should also emphasize the choice of an ensemble method for the glyph-based visualization design.

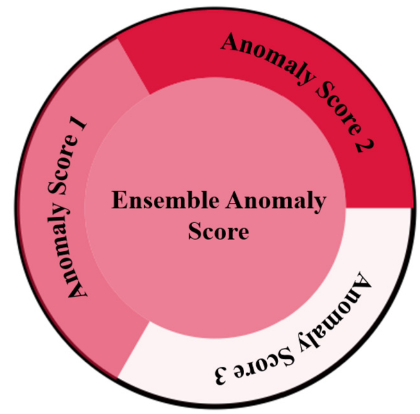

Fig. 2. The glyph visualizes three anomaly scores and their ensemble (aggregate). Anomaly Score 1 visualizes a value of 0.1, Anomaly Score 2 visualizes a value of 0.6, Anomaly Score 3 visualizes values of 1.0, whereas the equally weighted center visualized the Ensemble Anomaly Score of 0.6.

5.4. Feature-based anomaly score

The third anomaly score is motivated by engineers' typical approach to analyzing testbed data through temporal trend analysis (see Fig. 1). Some anomalies occur slowly over time and may be caused by the wearing of components. In contrast to point outliers [14], we consider that these anomalies may occur in a specific context and subsequence of the cycle, i.e., when the engine drives a particular engine speed. Considering this, we are interested in segmenting the time series of all cycles into equally sized windows. Further, we calculate for each time window and channel a generic set of domain-relevant time series features. In consultation with domain experts, we consider a specific set of features such as mean, minimum, and maximum to characterize the behavior of channels. However, the feature set is not complete and can be extended by additional or more complex features (i.e., variance, standard deviation, kurtosis [44]). Consequently, features and windows are calculated for a channel in all windows and are aggregated to a single dataset describing a channels' time series within one cycle. This method is then applied for each channel on either the reference and unseen data sets. Anomalies between features of the reference and unseen data sets are calculated by a Euclidean distance measure [45] and are mapped to a unified anomaly score as proposed in the methods above.

6. Visual encoding and considerations

This section explains how we use three anomaly scores for a glyph-based visualization. Also, an example of how to identify conspicuous channels within a cycle either in a matrix representation and a ranked channel list is given. The visual considerations explained in this section will be brought together in the prototype, describing the visual analytics approach by the prototype implementation.

6.1. Cycle anomaly glyph

The proposed glyph in Fig. 2 is flexible and independent of the underlying analytical methods for anomaly detection, as long as it implements the framework for calculating unified anomaly scores between 0 and 1 (Section 5.1). Anomaly scores are mapped to the outer circular segments of the glyph representation to color a color gradient ranging from white (0) to red (1). When designing the glyph, our aim was that no algorithm could detect all kinds of anomalies relevant for different applications. As a result, we choose an extensible glyph design, achieved by its circular shape, which offers the capability to add and remove anomaly scores in their according to circular segments. The glyph's main visual focus stays at the center circle, which represents an equally weighted average of anomaly scores combined, labeled as the ensemble anomaly score.

During our work, visualization experts' primary concern regarding the presented glyph design was the benefit compared to more straightforward visualizations, such as line plots. As stated in the design study, line plots are a well-known visualization type and comprehensible to the target group. However, our approach has advantages over line-plot-based visualizations. In general, testbed cycles as granularity level are highly comprehensible for engineers. Therefore, cycles are visualized as individual and complete entities, whereas the glyph design offers the following opportunities: (1) As a visual entity, it can be selected by users for further exploration, reasoning, and drill-down. Also, glyphs can be selected interactively to be defined as the reference for the underlying semi-supervised learning algorithms to identify contrasts between populations (**T1, T2**). (2) The glyph design can be extended with several anomaly detection algorithms by adding additional outer circular segments. (3) The glyph can be visualized on its own as a quick overview of an engine's condition. The idea is based on the idea of involved engineers having a simple "traffic light like" system, which also encouraged us to develop the presented glyph-based approach.

6.2. Identification of anomalous channels

As stated above, we choose three anomaly scores by their capability to further explore single anomalous channels. After an anomalous channel has been identified in the glyph representation, users are interested in the cause of that anomaly. Therefore, we visually represent anomalies for all anomaly scores, as follows:

Matrix-based identification of anomalous channels. The correlation deviation matrix calculated for the correlation-based anomaly score is shown in step (B2) of Fig. 3. Basically, in this symmetrical matrix, deviations of correlations of channels within a given cycle with respect to the selected reference cycle are visualized. More specifically, we compute the difference of the correlation matrices of these two cycles and show the result by color-coding the cells of the difference matrix. Hence, levels of red representing the anomaly score of channel correlations. This matrix representation supports the analysis goal to determine and quantify visual patterns for pattern-driven visual exploration. With appropriate matrix reordering methods, we can use this display to search for typical patterns in matrix visualizations, including line patterns and block patterns [46]. Most importantly, if one sensor shows an anomalous behavior, its correlation difference values to many or all other sensors will be rather large, leading to line patterns. Such visual patterns attract the attention of the analyst and are a starting point for drilling down into the respective sensor data (Fig. 3 (C2)).

Ranked channel list. The regression-based anomaly score and the feature-based anomaly score can be explored by the ranked channel list, as proposed in Fig. 3 (B1). Channels are ranked accordingly to the anomaly scores, calculated either through the mean average error (regression-based) or Euclidean distance measure (feature-based). Channels that deviate from the reference are listed and ranked by their anomaly score. This enables a guided approach for exploring anomalies and simplifies data analysis. By clicking on channel names, users can explore the reference and the anomalous channel time series by visual comparison in juxtaposition for hypothesis generation (Fig. 3 (C1)).

6.3. Prototype

The workflow of the approach, applied to data of the given use case, is exhibited and briefly described in Fig. 3. It shows screenshots of the implemented prototype, whereas further explanations are given in the following: In (A) glyphs are placed in a grid, with each cell representing a test cycle in chronological order (from top left to bottom right, inspired by the calendar-based view [47]). Note that the reference cycle is interactively selectable and represented by a white circle, as visible in the top left, or first, glyph. In (A), also three interaction possibilities are visible: (i) to add flexibility to the visual comparison of glyphs, the user can interactively change the anomaly score threshold to values between 50%–200% of the original value (ii) the user can change the number of displayed glyphs by filtering them by a "from–to" range slider (iii) glyphs can be filtered to visualize every x^{th} glyph only. Anomaly scores of interesting glyphs can be selected for further exploration by a drill-down in (B1) and (B2): In (B2) a drill-down example to inspect and identify one or more conspicuous channels within the selected cycle by a matrix representation visualizing the correlation-based anomaly score is shown. An example of a visual perceptive line pattern is outlined, representing a possibly conspicuous channel. Further, the conspicuous channel can be selected in the matrix for exploration and comparative analysis with the reference line plot in (C2). In (B1), an example of the ranked mean average error representation of the regression-based and feature-based anomaly scores is given, in which its drill-down capabilities are visible in (C1). In general, drill-down information needs to be investigated and interpreted by domain experts. However, our approach supports users in identifying interesting data by visually highlighting deviating cycles and sensors.

The prototype has been implemented using a Django Python web framework working in the backend and an Angular.js frontend. Also the prototype is depended on the cyclic data structure, which is described in Section 4. Originally the prototype was designed and implemented to calculate anomaly scores online after user interaction through the scikit-learn machine learning library [48]. Nevertheless, model training and calculations were too time-consuming causing a significant time delay for the user. Therefore we anticipate all possible user interactions, especially the reference cycle selection, and precalculate models and the associated anomaly scores. For the given dataset, the precalculation takes roughly four days on an average working notebook. As a side note, interactive line plots and heatmaps in the prototype have been created with the JavaScript visualization library Plotly.js [49] and are anonymized in Fig. 3, Fig. 5, Fig. 7 and Fig. 8 screenshots.

7. Evaluation

We conducted a qualitative pair analytics evaluation [9] with three subject matter experts (SME), who represent the target user group identified in Section 4.2, and the dataset described in Section 4.1. The main target was to evaluate either the comprehensibility of the different views and the underlying automated data analysis, along with the capabilities and limitations in supporting users with their daily condition monitoring analysis goals. According to the pair analytics protocol, the evaluation is done by a human-to-human interaction of one SME and one visual analytics expert (VAE), in which the SME acts as the navigator and the VAE as the driver (operator) of the visual analytics tool. In general, all three SME participants stated that the visual analytics tool could be of great benefit to support them in their daily work for two reasons: The visual analytics tool supports engineers in analyzing testbed data (1) more efficiently by highlighting interesting data on different granularity levels and (2) more effective by enabling the analysis of the entire dataset and not only a subset of

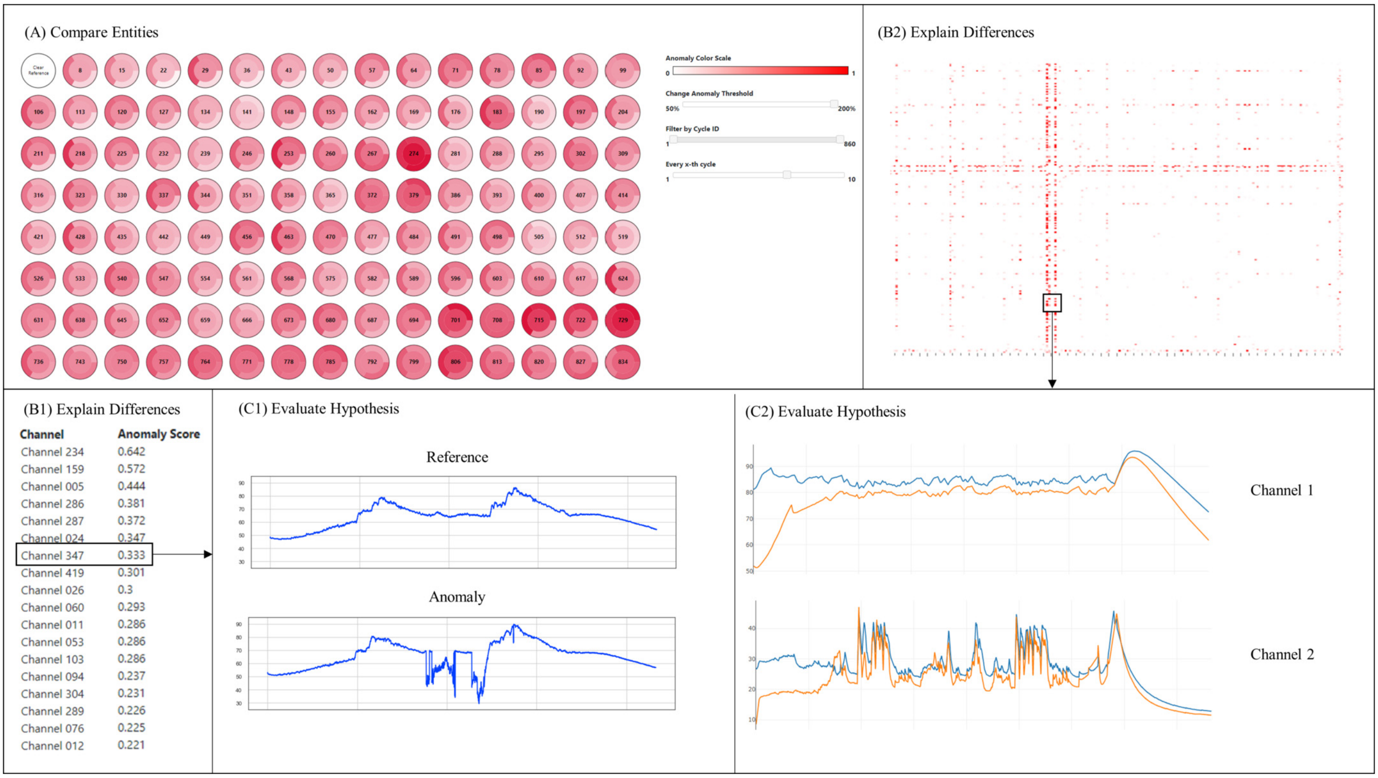

Fig. 3. (A) Left: Differences between the selected reference cycle and other cycles can be explored, whereas glyphs are positioned in a time-ordered grid. Right: Besides some filter capabilities, the anomaly score threshold can be interactively changed by a ruler. (B1) Anomalous channels found by the regression-based anomaly score can be explored by the ranked mean average error channel list. (B2) Anomalies found by the correlation-based anomaly score can be explored in the matrix representation (C) A hypothesis can be evaluated by comparing the channel time series in the reference cycle and the cycle of interest.

well-known channels. To give evidence to that statement, we connect participants comments and actions during the pair analytics evaluation to the task definition of Section 4.3 in the following:

Each evaluation session of two hours started with an introduction to the visual analytics approach and a short demonstration of the prototype. It is notable that all three participants (**P1**, **P2**, **P3**) gained a quick understanding of the concept for two reasons: First, we conducted the design study with the same engineers and connected findings of the study with explanations of our visual analytics tool. Second, the design study clearly identifies the tasks and goals of engineers. Therefore the visual analytics prototype accurately addresses the needs of engineers. In general, participants appreciated our effort in developing a decision support system supporting engineers in handling a large amount of data for their condition monitoring tasks.

The actual pair analytics evaluation sessions started by defining a reference cycle in the glyph-based overview (**T2**). **P1** and **P3** appreciate the capability of selecting the reference cycle interactively in the visualization. However, **P2** questioned the necessity of interactively selecting the reference cycle, because the testee is likely to be in good condition before the first cycle, considering that the testee runs through an extensive health check at the beginning of the entire durability test. We are aware of the fact that selecting the first cycle may be an appropriate default choice, but we wanted to keep the analysis more flexible. Nevertheless, all participants selected the first cycle as their reference.

After the reference cycle has been selected, other glyphs in the overview turned red regarding their anomaly scores (see Fig. 3 (A)). All participants were immediately curious about exploring these anomalies by the visualization and easily identified cycles that appear interesting to them (**T1**). **P2** and **P3** pointed at cycles that had a more intensive color of red than the majority of all cycles, whereas **P1** mentioned that all glyphs that visualize at least a small anomaly score are interesting. However, in a productive use scenario, the exploring strategy may differ since not all cycles are available from the beginning, and new data would be explored incrementally regularly as it becomes available. At this stage of the visual data analysis interesting data is visually highlighted, which enables the further exploration of anomalies in the succeeding views (**T5**).

T3 and **T4** are both achieved by exploring one anomaly scores of a specific cycle: (1) The correlation-based anomaly score and the correlation difference visualization in Fig. 3 (B2) were comprehensible to the participants as they were able to identify conspicuous channels. However, participants articulated the need for a more guided approach to engaging engineers using the matrix visualization because it appears overloaded and overwhelming to engineers. (2) The regression-based anomaly score was also highly comprehensible to participants since they have a general understanding of regression models. On the other hand, we avoided explaining the actual regression model Random Forest to participants in detail. In comparison to the correlation difference matrix, participants commented that the exploration of conspicuous channels is more accessible by the ranked channel list (Fig. 3 (B1)). Also, they expressed their interest in additional guided approaches in the other views, considering that such rankings represent an exact order on what channels to focus on, especially if they are short of time during their analysis.

As the last step of the visual analysis, participants evaluated the hypothesis of channels being anomalous by comparing anomalous line plots with their reference cycle equivalents (see Fig. 3 (C1 + C2)). From a data perspective, engineers approved that all explored anomalies are interesting because they highlight a significant difference to the reference. Some of the anomalies were interesting from a domain perspective, but others were explicable and irrelevant for the condition monitoring task. Another type of anomaly

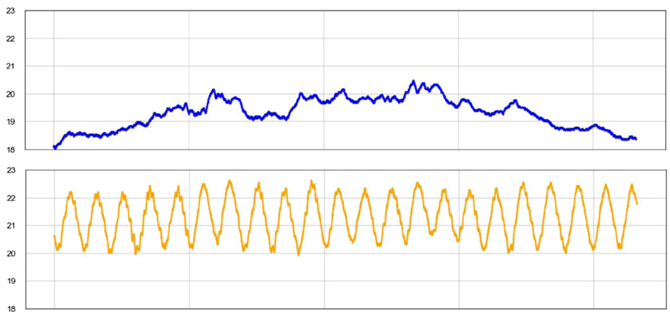

Fig. 4. A domain-irrelevant anomaly recorded by a temperature sensor in the engine.

that has been detected during evaluation is related to defective or unconnected sensors. Line plots of these anomalies visualize a constant or noisy signal. Therefore, we characterize three types of anomalies that have been found during evaluation: (1) Domain-irrelevant, (2) Domain-relevant, and (3) Defective sensors.

During evaluation data of a past 2,000 hours durability test was available. We managed to explore about one twentieth of the whole dataset and on average 30 anomalies in each evaluation session. It is worthwhile noting that in the first cycle, which all participants selected as the reference, two sensors were not connected to the system. Therefore, those sensors prominently showed deviations in all other cycles, which information is domain-irrelevant after the sensors got connected. This underlines the significance of the reference cycle selection along with the uncertainty of an industrial dataset. SMEs were able to confidently interpret 90% of the explored anomalies, whereas for the other 10% SMEs were enthusiastically using the prototype to further explore the temporal evolution of the anomaly in other cycles to further generate hypothesis.

Overall, we evaluated that the visual analytics prototype receives acceptance from all participants. They confirm the benefit of the proposed visual analytics approach and are interested in using the prototype in a production scenario. In the next section, we discuss the aspects and results of the evaluation in greater detail to motivate future and ongoing work in the subsequent sections.

8. Discussion

Our visual analytics approach has been designed and evaluated on testbed data, but we emphasize that it is not limited to the automotive domain. At least the glyph-based overview should be applicable to any other cyclic multivariate data set, as long as the underlying automated data analysis methods and dependent visualization techniques are adapted to the specific domain, if necessary. The visual analytics prototype has been proven to be useful for collaborators, as they clearly identified advantages in terms of efficiency and effectiveness compared to their current workflow. As a generic abstraction of anomalies in the glyph-based overview, the calculations of anomaly scores are exchangeable, and more extensive research on additional anomaly detection methods for the use case needs to be done. Our approach has been intended to be a generic solution for cyclic time series data. Nevertheless, developing anomaly detectors tailored to the domain-specific characteristics of data and possible anomalies could be of great benefit in any application domain.

The evaluation demonstrated that anomalies could be characterized in three manners, and we discuss examples of all of them, each detected by a different anomaly detector as follows:

(1) Domain-irrelevant: In a dynamic process like a durability test, with changing environmental variables throughout the seasons, many anomalies can occur from a data perspective. In contrast, from a domain perspective, many of these anomalies are plausible and, therefore, domain-irrelevant for the given condition monitoring task. For instance, in Fig. 4 an anomaly, which has been explored during the evaluation, is illustrated by two line plots. The upper line plot visualizes the reference, and the lower line plot visualizes the suspicious time series of a sensor measuring a relevant airflow temperature in the engine. From a data perspective, the suspicious time series sequence can be interpreted as an anomaly because of the evident differences. Nevertheless, to clarify the example further, the anomaly has been recorded four months after the reference. The anomaly simply illustrates the periodic oscillation of the heating control, which was turned on during the test and, therefore, also influenced the testbed data. This anomaly has been detected through the correlation-based anomaly score. Pairwise relations between temperature channels often linearly correlate, which lies in the nature of physics. However, if a single temperature channel deviates in its behavior like in the given an example, it is visually recognizable in the correlation deviation matrix as proposed in Section 6.

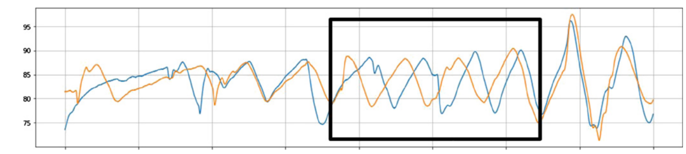

Fig. 5. Comparison of reference (blue) and anomalous time series (orange) in superposition shows a reflection phase change in the highlighted subsequence. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

(2) Domain-relevant: Returning the example in Section 3, we now discuss how this failure could be detected through our visual analytics approach. Considering that the temperature increase, which caused the failure, appeared only in a specific context, the failure could have been detected through the similarity-based approach. The temporal binning applied in that approach adds sensitivity for the detection of subsequence outliers [14]. It supports the detection of the relevant subsequent time series indicating the problem related to our example. For the use case, in particular, the method may profit from a domain-knowledge-based binning technique to examine specific domain-relevant operating modes separately, if possible. However, we present a generic approach, and it is up to the domain and use case what binning technique to apply.

The channels in our dataset are strongly related with each other and represent the underlying physics of a combustion engine. Therefore anomalies and misbehavior of components in the engine may be detected in several channel time series. Regarding our example, we first could detect the problem through another sensor, measuring the temperature of the liquid that is responsible for cooling the critical component (see Fig. 1). The anomaly can be recognized as a reflection phase change in the temperature of that liquid in a specific subsequence of the cycle compared to the reference (Fig. 5). It indicates a problem with the liquid that, in the end, caused the failure.

(3) Defective sensors: The third category of anomalies that have been detected during the evaluation is defective sensors. These anomalies can occur in the whole test cycle and may be recognized by implausible or constant values. Nevertheless, they also occur in subsequences of a cycle because the malfunction appears occasionally. One example that has been detected through the regression-model-based anomaly detector is illustrated in Fig. 3 (C1): The figure illustrates an anomalous temperature time series, which shows similar behavior to the reference. However, temperature drops, which are not plausible from a domain point of view, indicating that the sensor is defective and needs to be replaced.

To conclude the evaluation and the discussion, we want to emphasize that domain expertise is essential for our dataset's condi-

Table 2
Anomaly knowledge-base example records. User-labels are stored in a database representing data to recall anomalies for further automated processing.

Id	Description	Channel	Cycle id	Ref id	Start	End	Range start	Range end	Class
1	suspicious temperature drop	Channel 43	116	1	08:34	09:23	15,54	67,43	Drop
2	shift in whole time series	Channel 12	194	1	00:00	16:00	32,32	49,34	Shift
3	shift at the end of the test	Channel 2	276	1	14:34	16:00	101,31	105,56	Shift
4	unexpected peak	Channel 80	403	1	03:43	03:56	93,95	104,66	Peak

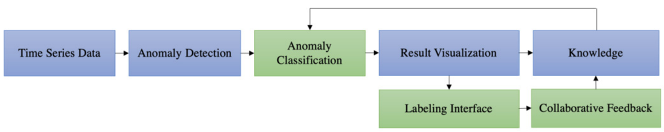

Fig. 6. The current workflow (blue) is extended by a visual interactive labeling interface, collaborative feedback and anomaly classification (green). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

tion monitoring goals. From a data point of view, many anomalies occur naturally on automotive testbeds and similar datasets. Extracting valuable knowledge depends on both the extensive experience and know-how of engineers. In the next section, we process our findings and propose a visual interactive labeling interface for anomaly classification.

9. Visual interactive labeling and anomaly classification

The visual interactive labeling (VIAL) interface is a collaborative tool that enables the capturing of domain knowledge for verification support as an extension of the presented approach. Our conceptual workflow process, from Fig. 6, derives from the VIAL process by Bernard et al. [50] and illustrates the current process (blue) and the visual interactive labeling and classification extension (green). The extension mainly consists of three procedures that are further explained in the following subsections: the visual interactive labeling interface, collaborative feedback and anomaly classification.

9.1. Visual interactive labeling interface

In the last step of the analysis process in our prototype, users compare line plots of the anomalous time series to its reference. They interpret the anomaly by their domain knowledge and should be able to store their findings for three purposes: (1) record findings for future reference, (2) classification of anomalies in unseen datasets and cycles, and (3) request collaborative feedback of domain experts. The evaluation showed, that the diversity and large number of recurring anomalies in testbed data require an anomaly classification. Users should be able to define domain-related anomaly classes themselves. Also, users should be able to select a subsequence in the time series that they interpret as an anomaly and additionally assign a class and a textual description of their interpretation. It is essential to consider that putting an anomaly into the context of the references facilitates the reasoning for it. A first version of the labeling interface has been implemented and is illustrated in Fig. 7. A label stores several metadata of the anomaly (i.e., cycle numbers of the reference and the anomaly, channel names), a user-selected subsequence of the time series, a user-defined anomaly class, and a textual description of the anomaly. The provided metadata offers starting points for further analysis, e.g., categorization of lables by failure type, or automated linking to stored test reports from previous cycles. The more data is collected in the anomaly knowledge-base, the more opportunities to cross-link and compare anomalies are given.

9.2. Collaborative feedback

The collaborative feedback allows additional domain experts to explore and refine anomaly labels. Especially if the user has questions about the anomaly, additional feedback can improve the anomaly knowledge-base quality. Further, users can add self-defined classes, which have significant influence on the automated anomaly classification. Hence, the quality of these classes should be improved using the feedback of multiple domain experts.

9.3. Anomaly classification

A challenge we tackle in our ongoing work is the classification of anomalous time series by using the anomaly knowledge-base. We explicitly focus on a generic approach, which is not necessarily attached to the automotive domain. Anomalies in our work often occur in the context of their reference. Therefore, the anomalous time series, as well as the difference between the anomalous and reference time series need to be taken into account for a classification. In the first step an abstract description of time series labels for classification task needs to be given. Consequently, we normalize time series labels and extract a set of generic time series features from both the anomalous time series and the difference time series between the anomaly and the reference. The idea behind this approach is that time series anomaly classes can be either described as a univariate or as bivariate in the context of the reference. Therefore, we extract the following time series features from the anomalous time series, as well from the difference time series: mean, minimum, maximum, median, kurtosis, skewness, number of peaks, standard deviation, absolute sum of changes and length.

As a generic approach, our system has no foreknowledge or descriptions about anomaly classes, which are entirely user-label-defined. Therefore, we explored the given dataset and experimentally defined five time series anomaly classes that may be interesting in time series anomaly classification tasks: (1) Drop (2) Shift (3) Peak and (4) Oscillation. Since the anomaly classifier should also be able to distinguish between anomalous and normal time series, we also defined a class (5) No Anomaly. In the next step, we further explored the automotive dataset and labeled 40 anomalies accordingly to the five self-defined anomaly classes as a training data set. Example labels from the anomaly knowledge-base are illustrated in Table 2. Consequently, we extracted the generic time series features for the labels and trained a random forest classifier [42]. The training error of the model was zero, which is an indicator that the generic time series feature set may give a sufficient description of the anomaly classes. To experimentally investigate the performance of the classifier, we tested the model on unseen anomalies. The results look promising, considering that the system has no foreknowledge about anomaly classes, which are described by generic time series features and a small user-label training data set. Two examples are shown in Fig. 8, whereas we added the model confidence by the prediction probability for each class.

Fig. 7. Labeling interface for building an anomaly knowledge-base and for classification. (A) The user can interactively select a subsequence of the anomalous time series by mouse drag. The selection is linked and synchronizes with the reference time series to visually aid the comparison. (B) Metadata from the dataset is automatically filled in the input fields of the user interface (gray background color) (C) Users can select an existing category / class for the anomaly or create a new one. Also, a textual description of the anomaly must be given.

Those preliminary results open up many directions for future work and are discussed, among others, in the next section.

10. Future work

The main challenge in the presented dataset is the massive amount of occurring anomalies. Exploring them can be very time consuming, even if our prototype facilitates the anomaly exploration process. It is also notable that the same durability test is often conducted on multiple identical testees in parallel, which enables a more comprehensive analysis. On the one hand, the analysis of numerous testees would increase the size of the dataset even more, but on the other hand, it allows exchanging knowledge among them. To take this one step further, we address preserving knowledge of past durability tests to analyze novel engines and present tests. Therefore, we are currently investigating the automated classification of anomalies through the user-centered interactive labeling anomaly knowledge-base. To make data analysis more efficient, we also consider guided approaches for future work. This is related to the classification of anomalies through the anomaly knowledge-base. Additionally, research on the reasoning of anomalies needs to be conducted, whereas we plan to focus on visual causality analysis [51] to explore the root cause of an anomalous time series. In the next sections, we describe the focus of our ongoing work and future work.

10.1. Visual interactive labeling

Preliminary results on the interactive labeling of anomalies open up promising directions for future work. We hope that further investigations reveal that anomalies can be detected and classified into user-label-trained classes. As the next step, we propose the automated classification of anomalies to reduce workload of users by setting their focus on unknown not classifiable anomalies. Some time series anomalies occur in the context of multiple variables. Therefore, we also suggest to take multivariate time series labeling into account to make the subsequent classification more comprehensive to the user.

10.2. Guidance

Domain experts can use our tool to interactively label anomalies that have either been found automatically by the system or detected by the users due to their domain knowledge. This is an important functionality of anomaly detection systems, as the detection may include false positives, or the severity of detection may change over time. Such labels are saved and collected in an anomaly knowledge-base. In the future, we will develop a self-learning system that learns from the collected labels to automatically identify and classify unknown use-case, situation- and context-specific anomalies. From a growing set of anomalies, learning methods will be able to generalize the labeled patterns, in-

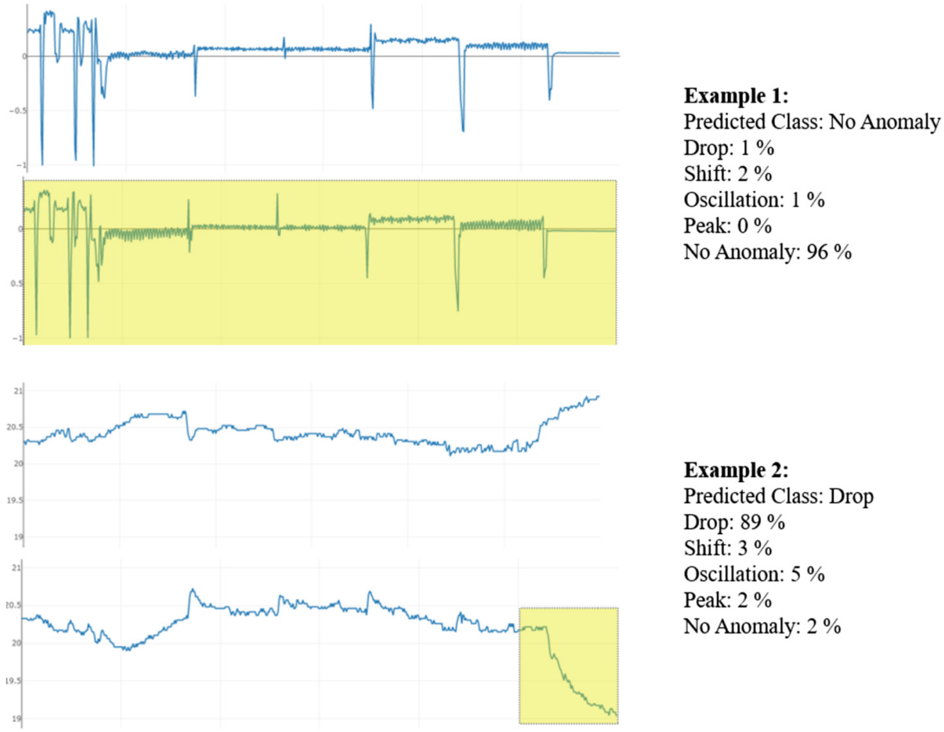

Fig. 8. Two time series have been classified by a user-label trained classifier. Both examples show a time series (bottom) and its reference (top). The yellow background is the subsequence that we interactively selected in the prototype, whereas the classifier results on the selection are illustrated on the right. Example 1 shows pressure measurements, which time series is very similar to its reference. The classifier correctly classified the time series as "No Anomaly". Example 2 shows a temperature times serie, which suspiciously drops at the end in comparison to its reference. The classifier classified this time series correctly as "Drop".

cluding the domain knowledge, and hence support more efficient anomaly detection and less user-label requirements for future operations. An important part of the system will be to provide techniques (e.g., visualizations) and guidance approaches [52] to support the users to identify and understand the factors producing the anomalies (cf., root-cause analysis) and to compare the possible change of anomalies over time.

10.3. Visual-aided causal discovery

Future extension of the current work is to go beyond mere correlations and engage causal discovery methods [53] to be able to recover the true root-cause of triggered anomalies via advanced visual analytics techniques. Such novel approaches can facilitate more robust and reliable anomaly detection methods, as domain knowledge can be seamlessly injected into the causal models with various interactive visualization tools. Existing works by Holst et al. [54] or Wang and Mueller [51] incorporate causality with interactive visualization tools that demonstrate promising results in this direction. However, the former methods are not taking into account time series data, which causality plays a significant role since, in the context of time *the cause must always precede the effect*. For future work, we aim at addressing the above challenges.

10.4. Scalability

The current state of the proposed system lacks maturity in terms of scalability in different views, which we address for future work. The glyph-based overview (see Fig. 3 (A)) does not scale well for many cycles. Therefore, future work will concentrate on advanced visual aggregation and composition techniques [55] for the glyph-based overview. Another occasion to improve scalability would be clustering and visual evolution tracking, as proposed by [56]. In comparison to the current grid-based system, such techniques could be of great benefit to the user. It also may support the user applying a data-driven way to choose reference data. For future work, we also address the scalability and usability of the matrix view. For example, a Magnostics approach [57] to search and rank patterns in the correlation deviation matrix, combined with other interactions, would reduce complexity and support users in exploring interesting channels. The glyph design has been developed on a small set of anomaly detectors. Therefore, another opportunity for future work is the investigation of additional anomaly detectors, but also the limitations of the current glyph design in terms of the scalibility.

10.5. Evaluation

The conducted prototype evaluation focused on the comprehensibility of different views and the underlying data analysis, but did not quantify the performance of our prototype. To further our research we intend to perform quantitative evaluations of the prototype regarding the anomaly detection and exploration, but also of the ongoing work on anomaly classification through user-label-trained classifiers. The investigated industrial dataset is unsuitable for a quantitative study by reason of missing labels and uncertainty in the data. Therefore we plan to evaluate with a public dataset.

11. Conclusion

In conclusion, we presented a glyph-based visual analytics approach visualizing an ensemble of several anomaly detectors. It has been designed and applied to automotive testbed data and enables the visual exploration of anomalies of cyclic multivariate time series data. The prototype evaluation results are promising, and we have found ways to potentially increase the effectiveness and efficiency of the domain experts' data analysis. Present findings revealed that from a data perspective, many anomalies occur on the investigated dataset. As a consequence, we provided detailed concepts and preliminary results on the continuation of the presented work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been supported by the FFG, Contract No. 854184: “Pro^{2}Future is funded within the Austrian COMET Program Competence Centers for Excellent Technologies under the auspices of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs and of the Provinces of Upper Austria and Styria. COMET is managed by the Austrian Research Promotion Agency FFG.”

We thank the reviewers whose comments and suggestions helped improve and clarify earlier drafts of the manuscript.

References

[1] J. Suschnigg, B. Mutlu, A. Fuchs, V. Sabol, S. Thalmann, T. Schreck, Exploration of anomalies in cyclic multivariate industrial time series data for condition monitoring, in: BigVis 2020 - 3rd International Workshop on Big Data Visual Exploration and Analytics, in: CEUR Workshop Proceedings, vol. 2578, 2020, pp. 1–8.
[2] G. Vial, Understanding digital transformation: a review and a research agenda, J. Strateg. Inf. Syst. 28 (2) (2019) 118–144.
[3] J.P. Zeiringer, S. Thalmann, Knowledge risks in digital supply chains: a literature review, in: 15th International Conference on Wirtschaftsinformatik, 2020, pp. 1–16.
[4] K. Vogelsang, K. Liere-Netheler, S. Packmohr, U. Hoppe, Success factors for fostering a digital transformation in manufacturing companies, J. Enterp. Transf. (2019) 1–22.
[5] S. Thalmann, J. Mangler, T. Schreck, C. Huemer, M. Streit, F. Pauker, G. Weichhart, S. Schulte, C. Kittl, C. Pollak, M. Vukovic, G. Kappel, M. Gashi, S. Rinderle-Ma, J. Suschnigg, N. Jekic, S.N. Lindstaedt, Data analytics for industrial process improvement a vision paper, in: IEEE 20th Conference on Business Informatics (CBI), Vol. 02, 2018, pp. 92–96.
[6] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, C. Tominski, Characterizing guidance in visual analytics, IEEE Trans. Vis. Comput. Graph. 23 (2017) 111–120.
[7] R. Elshawi, S. Sakr, D. Talia, P. Trunfio, Big data systems meet machine learning challenges: towards big data science as a service, Big Data Res. 14 (2018) 1–11.
[8] G. Andrienko, N. Andrienko, S. Drucker, J.-D. Fekete, D. Fisher, S. Idreos, T. Kraska, G. Li, K.-L. Ma, J. Mackinlay, et al., Big data visualization and analytics: future research challenges and emerging applications, in: BigVis 2020 - 3rd International Workshop on Big Data Visual Exploration and Analytics, in: CEUR Workshop Proceedings, vol. 2578, 2020, pp. 1–8.
[9] R. Arias-Hernandez, L.T. Kaastra, T.M. Green, B. Fisher, Pair analytics: capturing reasoning processes in collaborative visual analytics, in: 44th Hawaii International Conference on System Sciences, IEEE, 2011, pp. 1–10.
[10] A. Maier, T. Tack, O. Niggemann, Visual anomaly detection in production plants, in: Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics, SciTePress, 2012, pp. 67–75.
[11] S. Agrawal, J. Agrawal, Survey on Anomaly Detection Using Data Mining Techniques, Procedia Computer Science, vol. 60, Elsevier, 2015, pp. 708–713.
[12] M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Process. 99 (2014) 215–249.
[13] G. Blanchard, G. Lee, C. Scott, Semi-supervised novelty detection, J. Mach. Learn. Res. 11 (2010) 2973–3009.
[14] M. Gupta, J. Gao, C.C. Aggarwal, J. Han, Outlier detection for temporal data: a survey, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 2250–2267.
[15] C. Gutschi, N. Furian, J. Suschnigg, D. Neubacher, S. Voessner, Log-based predictive maintenance in discrete parts manufacturing, in: 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Vol. 79, Elsevier, 2019, pp. 528–533.
[16] P. Zhao, M. Kurihara, J. Tanaka, T. Noda, S. Chikuma, T. Suzuki, Advanced correlation-based anomaly detection method for predictive maintenance, in: IEEE International Conference on Prognostics and Health Management, 2017, pp. 78–83.
[17] M. Shcherbakov, A. Brebels, N. Shcherbakova, V. Kamaev, O. Gerget, D. Devyatykh, Outlier detection and classification in sensor data streams for proactive decision support systems, J. Phys. Conf. Ser. 803 (2017).
[18] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. 41 (3) (2009) 1–58.
[19] G.A. Susto, A. Cenedese, M. Terzi, Time-series classification methods: review and applications to power systems data, in: R. Arghandeh, Y. Zhou (Eds.), Big Data Application in Power Systems, Elsevier, 2018, pp. 179–220.
[20] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, L. Dong, Detecting anomalies in time series data via a meta-feature based approach, IEEE Access 6 (2018) 27760–27776.
[21] F. Zhou, X. Lin, C. Liu, Y. Zhao, P. Xu, L. Ren, T. Xue, L. Ren, A survey of visualization for smart manufacturing, J. Vis. 22 (2019) 419–435.
[22] H. Janetzko, F. Stoffel, S. Mittelstädt, D.A. Keim, Anomaly detection for visual analytics of power consumption data, Comput. Graph. 38 (2014) 27–37.
[23] D. Keim, H.-P. Kriegel, M. Ankerst, Recursive pattern: a technique for visualizing very large amounts of data, in: Proceedings of the IEEE Visualization Conference, 1995, pp. 279–286.
[24] M. Weber, M. Alexa, W. Müller, Visualizing time-series on spirals, in: IEEE Symposium on Information Visualization, 2001, 2001, pp. 7–13.
[25] W. Wu, Y. Zheng, K. Chen, X. Wang, N. Cao, A visual analytics approach for equipment condition monitoring in smart factories of process industry, in: IEEE Pacific Visualization Symposium, 2018, pp. 140–149.
[26] P. Buono, C. Plaisant, A. Simeone, A. Aris, B. Shneiderman, G. Shmueli, W. Jank, Similarity-based forecasting with simultaneous previews: a river plot interface for time series forecasting, in: 11th International Conference Information Visualization, IEEE, 2007, pp. 191–196.
[27] J. Xia, F. Ye, W. Chen, Y. Wang, W. Chen, Y. Ma, A.K. Tung, LDSScanner: exploratory analysis of low-dimensional structures in high-dimensional datasets, IEEE Trans. Vis. Comput. Graph. 24 (2018) 236–245.
[28] K. Xu, M. Xia, X. Mu, Y. Wang, N. Cao, Ensemblelens: ensemble-based visual exploration of anomaly detection algorithms with multidimensional data, IEEE Trans. Vis. Comput. Graph. 25 (2019) 109–119.
[29] R. Borgo, J. Kehrer, D.H.S. Chung, E. Maguire, R.S. Laramee, H. Hauser, M. Ward, M. Chen, Glyph-based visualization: foundations, design guidelines, techniques and applications, in: Eurographics 2013 - State of the Art Reports, 2013, pp. 39–63.
[30] T. Ropinski, S. Oeltze, B. Preim, Survey of glyph-based visualization techniques for spatial multivariate medical data, Comput. Graph. 35 (2011) 392–401.
[31] J. Fuchs, P. Isenberg, A. Bezerianos, D. Keim, A systematic review of experimental studies on data glyphs, IEEE Trans. Vis. Comput. Graph. 23 (2017) 1863–1879.
[32] J. Fuchs, F. Fischer, F. Mansmann, E. Bertini, P. Isenberg, Evaluation of alternative glyph designs for time series data in a small multiple setting, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13, ACM, 2013, pp. 3237–3246.
[33] M.O. Ward, B.N. Lipchak, A visualization tool for exploratory analysis of cyclic multivariate data, Metrika 51 (2000) 27–37.
[34] N. Cao, C. Shi, S. Lin, J. Lu, Y.R. Lin, C.Y. Lin, TargetVue: visual analysis of anomalous user behaviors in online communication systems, IEEE Trans. Vis. Comput. Graph. 22 (2016) 280–289.
[35] E. Armengaud, Industry 4.0 as digitalization over the entire product lifecycle: opportunities in the automotive domain, in: Systems, Software and Services Process Improvement, 24th European Conference, EuroSPI 2017, Springer, 2017, pp. 334–351.
[36] S. Thalmann, G. Heimo, J. Suschnigg, M. Gashi, H. Ennsbrunner, A. Fuchs, T. Schreck, B. Mutlu, J. Mangler, G. Kappl, C. Huemer, S. Lindstaedt, Cognitive decision support for industrial product life cycles: a position paper, vol. 11, in: Cognitive 2019, IARIA, 2019, pp. 3–9.
[37] Q. Xin, 2-durability and reliability in diesel engine system design, in: Q. Xin (Ed.), Diesel Engine System Design, Woodhead Publishing, 2013, pp. 113–202.
[38] S. Miksch, W. Aigner, A matter of time: applying a data–users–tasks design triangle to visual analytics of time-oriented data, Comput. Graph. 38 (2014) 286–290.
[39] M. Sedlmair, M. Meyer, T. Munzner, Design study methodology: reflections from the trenches and the stacks, IEEE Trans. Vis. Comput. Graph. 18 (12) (2012) 2431–2440.
[40] H. Lam, M. Tory, T. Munzner, Bridging from goals to tasks with design study analysis reports, IEEE Trans. Vis. Comput. Graph. 24 (2018) 435–445.
[41] M. Gleicher, Considerations for visualizing comparison, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 413–423.
[42] L. Breiman, Random Forrest, Mach. Learn. 45 (2001) 5–32.
[43] X. Zhao, Y. Wu, D.L. Lee, W. Cui, iforest: interpreting random forests via visual analytics, IEEE Trans. Vis. Comput. Graph. 25 (1) (2018) 407–416.
[44] T.W. Rauber, F. de Assis Boldt, F.M. Varejão, Heterogeneous feature models and feature selection applied to bearing fault diagnosis, IEEE Trans. Ind. Electron. 62 (1) (2015) 637–646.
[45] S. Kloiber, J. Suschnigg, V. Settgast, C. Schinko, M. Weinzerl, T. Schreck, R. Preiner, Immersive analytics of anomalies in multivariate time series data with-proxy interaction, in: 2020 International Conference on Cyberworlds, CW 2020, IEEE, 2020, pp. 1–8.

[46] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, J.-D. Fekete, Matrix reordering methods for table and network visualization, Comput. Graph. Forum 35 (3) (2016) 693–716.
[47] J.J. Van Wijk, E.R. Van Selow, Cluster and calendar based visualization of time series data, in: Proceedings IEEE Symposium on Information Visualization, 1999, pp. 4–9.
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[49] Plotly Technologies Inc., Collaborative data science, https://plot.ly, 2015.
[50] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, M. Sedlmair, Comparing visual-interactive labeling with active learning: an experimental study, IEEE Trans. Vis. Comput. Graph. 24 (1) (2018) 298–308.
[51] J. Wang, K. Mueller, Visual causality analysis made practical, in: 2017 IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE, 2017, pp. 151–161.
[52] D. Ceneda, N. Andrienko, G. Andrienko, T. Gschwandtner, S. Miksch, N. Piccolotto, T. Schreck, M. Streit, J. Suschnigg, C. Tominski, Guide me in analysis: a framework for guidance designers, Comput. Graph. Forum 39 (6) (2020) 269–288.
[53] P. Spirtes, K. Zhang, Causal Discovery and Inference: Concepts and Recent Methodological Advances, Applied Informatics, vol. 3, Springer, 2016, p. 3.
[54] A. Holst, S. Pashami, J. Bae, Incremental causal discovery and visualization, in: Proceedings of the Workshop on Interactive Data Mining, 2019, pp. 1–6.
[55] J. Wang, S. Hazarika, C. Li, H. Shen, Visualization and visual analysis of ensemble data: a survey, IEEE Trans. Vis. Comput. Graph. 25 (9) (2019) 2853–2872.
[56] B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, P. Dragicevic, Time curves: folding time to visualize patterns of temporal evolution in data, IEEE Trans. Vis. Comput. Graph. 22 (1) (2016) 559–568.
[57] M. Behrisch, B. Bach, M. Hund, M. Delz, L. von Rüden, J.-D. Fekete, T. Schreck, Magnostics: image-based search of interesting matrix views for guided network exploration, Trans. Vis. Comput. Graph. 23 (1) (2017) 31–40.