DigiEMine: Towards Leveraging Decision Mining and Context Data for Quality Control

Beate Wais[1,2] and Stefanie Rinderle-Ma[3]

[1] University of Vienna, Faculty of Computer Science, Research Group Workflow Systems and Technology, Vienna, Austria
[2] University of Vienna, UniVie Doctoral School Computer Science DoCS, Vienna, Austria beate.wais@univie.ac.at
[3] Technical University of Munich, TUM School of Computation, Information and Technology, Garching, Germany, stefanie.rinderle-ma@tum.de

Abstract. Quality control processes in manufacturing often still rely on manual tasks. Applying decision mining can support users by providing valuable insight into the process. This paper discusses the potential of integrating contextual information into decision mining to achieve accurate and meaningful decision rules in the context of a case study stemming from the manufacturing domain. To explore this, a new approach, DigiEMine, is presented, which addresses the gap between information extraction and practical decision mining applications by integrating information extracted from engineering drawings with time sequence data in the form of diameter measurements of workpieces. The discovery of relational decision rules is enabled, allowing for contextualization of the decision rules. The output of this approach is presented in both textual decision rules and visually on engineering drawings, empowering users to make informed quality control decisions. The case study includes three datasets originating from cylindrical workpiece production. Results demonstrate the feasibility of the approach and the ability to generate meaningful decision rules across the tested datasets. Its potential applicability extends beyond the presented case study, with conceivable scenarios in multiple domains, such as healthcare or logistics, where integrating context information, such as regulatory data, with time sequence data is required to provide additional context for decisions.

Keywords: Decision Mining · Context Data · Manufacturing · Quality Control.

1 Introduction

Process mining, including process discovery, conformance checking, and process enhancement [1], plays an essential role in driving automation and digitalization and can be applied in multiple ways, delivering valuable insights into operations and enabling the identification of bottlenecks, inefficiencies, and deviations from the intended process flow. This information can be used to optimize production processes, reduce waste, and improve overall productivity [20]. An essential aspect of process mining involves decision mining, i.e., discovering decision

points and the underlying decision rules in processes [15]. The need for improving knowledge about and around decisions is increasing as *"[e]ffective decision making – that is connected, contextual and continuous – results in a host of business benefits, including greater transparency, accuracy, scalability and speed"* [4]. Decision mining enables increased transparency in processes by capturing the underlying logic of decisions, allowing users to understand the decisions in a process. Decision mining typically employs classification techniques and aims to provide decision rules that are in human-readable form. This, in turn, can lead to faster detection of deviations and allows for evaluation whether these detected deviations are intentional or due to errors, decreasing the time until errors are detected and thereby minimizing the impact of errors on the overall outcome.

Typically, the input data for decision mining comprises process event log data for determining decision points in the process and process data such as patient age or the loan amount to determine the decision rules at the decision points based on classification techniques, mostly decision trees [15]. In domains where IoT data provides context to process event data, e.g., manufacturing, logistics, and healthcare, sensor data might also influence decisions and should hence be part of potentially more complex decision rules, i.e., turning from, e.g., "temperature > 30" to "temperature exceeds 30 for three times in a row" [2,9,23].

Input data for decision mining might comprise additional structured or unstructured context data. Context data defined as being *"additional process-related information"* [5] might be crucial for decisions in a process. An example of context data in manufacturing are *engineering drawings (EDs)*. EDs are the source of information on how a product is going to be produced and also serve as input for quality checks after production [21] and, therefore, provide important process-related information.

Including context data explicitly in decision mining can lead to more accurate and meaningful results with decision rules that are set in the appropriate context. This means that the resulting decision model and the mined decision rule are more meaningful to employees using and interpreting the decision rules. However, integrating data, specifically less structured data such as images, is not trivial and might lead to features that are not easy to interpret for humans. Including context data explicitly, therefore, requires the use of features that can transport as much information as possible to the users. This can be done by building relational features, where two features are connected, e.g., "$age\_customer <= maximum\_age$". Multiple decision mining approaches exist in the literature; see [15], including approaches that enable the extraction of relational decision rules [3,14,22]. So far, to the best of our knowledge, no approach exists that enables the integration of time sequence data and unstructured context data. However, combinations of time sequence data and additional context data occur in multiple domains, for example in manufacturing where sensor data is set in relation to specifications. This paper, therefore, explores the integration of context data in decision mining embedded in a case study from the manufacturing domain to answer the following research question RQ:

[4] www.gartner.com/smarterwithgartner/how-to-make-better-business-decisions

RQ: How can context data, such as dimensioning information, and time sequence data be combined and integrated into decision mining algorithms?

We employ a case study methodology [19] to gain an in-depth understanding of the challenges and complexities of a specific use case and the corresponding implementation, providing valuable insights into its functionality and potential challenges. The main contribution of this paper is the introduction of the DigiEMine approach. This novel approach integrates unstructured context data with time sequence data for decision mining, allowing the construction of relational decision rules that provide explicit reference to the input data. Thereby, the traceability and reconfigurability of the resulting decision rules are increased. Traceability refers to understanding why a specific value is essential in a decision rule. In contrast, reconfigurability refers to decision rules being easily adapted if the underlying decision logic changes. The presented approach bridges the gap between information extraction from engineering drawings and its practical application in decision mining, contributing to a more seamless and effective automated quality control process.

The rest of the paper is organized as follows: a case study exploring the research challenges in depth is presented in Sect. 2. The DigiEMine approach is described in Sect. 3 and the results of applying the approach to the cases are presented in Sect. 4. The results are then discussed in Sect. 5 and related work is presented in Sect. 6. A conclusion is given in Sect. 7.

2 Case Study

The case study stems from the manufacturing domain, particularly the production of cylindrical workpieces, such as valve lifters. These workpieces are produced in small batches using a turning machine. The dimensions stem from a CAD (Computer Aided Design) model, which is nowadays mainly used in production. In addition, an engineering drawing is generated from the CAD model, where additional information, such as applicable regulatory guidelines and default tolerances, is noted. After producing the workpiece, its quality is assessed by measuring different attributes and comparing the measurements to the requirements specified in the corresponding ED. The best-case scenario would involve all specifications being part of the CAD model, including tolerances, which can be automatically extracted for quality control. However, engineering drawings are still frequently applied as a contractual basis and as a reference for quality control as the necessary information is often missing in the CAD models [12].

As shown in Fig. 1, the quality control process involves taking two measurements for efficiency and quality reasons. Firstly, a silhouette measuring machine (Keyence) checks the workpiece diameter. This step takes a few seconds but can be inaccurate as not all essential quality factors can be measured this way.

Therefore, the workpieces are transferred to a second measuring machine (MicroVu) to measure more attributes, e.g., surface quality and flatness, resulting in more precise results. This step takes a couple of minutes. Hence, an optimization of the quality control would be to classify instances as "ok" or "not ok" after Keyence and let only workpieces with a high probability of being "ok" continue to MicroVu. This optimization can be expressed by a decision point (DP1), highlighted by a red circle in Fig 1; at this point, the Keyence measurements should be compared to the dimensions and tolerances stated in the ED.

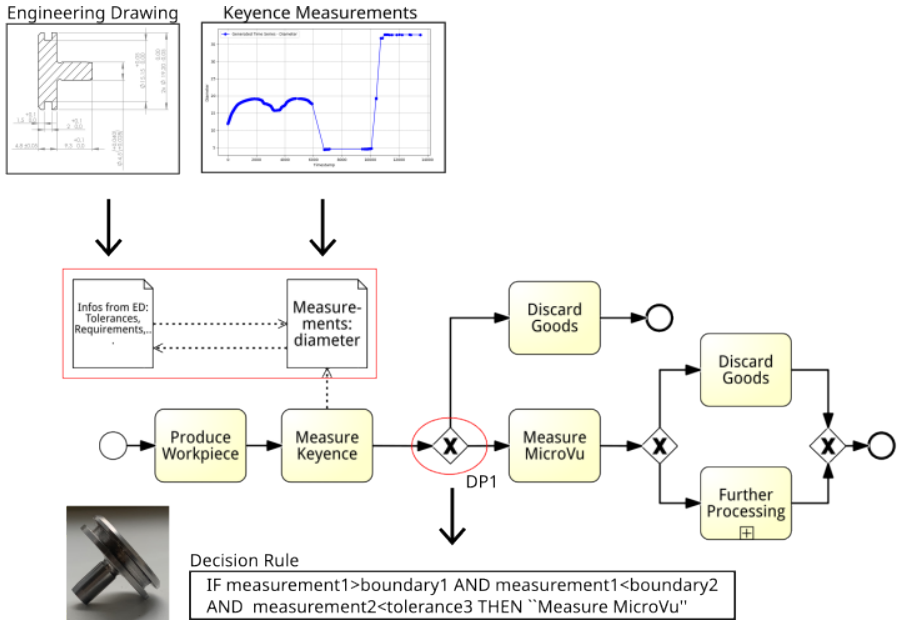

Fig. 1. Valve lifter production process, engineering drawings, measurement values, and resulting decision rule.

The Keyence measuring machine[5] works by illuminating the workpieces with a green LED and a telecentric lens. When the workpiece is put through the machine, it breaks the beam, creating a shadow on the sensor. Different features, such as size and angle, can be calculated by measuring this shadow. As the valve lifter is a cylindrical workpiece, the main feature is the diameter of the workpiece. The resulting measurements correspond to the outline of the workpiece (cf. close-up in Fig. 2). The data points up to timestamp *10000* (measured in milliseconds) correspond to the actual silhouette of the workpiece. For the remaining time, the measured values, including the steep increase, are artifacts produced by the robot

[5] https://www.keyence.eu/ss/products/measure/measurement_library/type/optical, accessed:12/04/2024

arm holding the workpiece in place while it is being measured. It can be seen that the measurements do not explicitly correspond to discrete values measuring each dimension but are continuous measurements, i.e., time sequence measurements, as the workpiece is pulled through the laser beam.

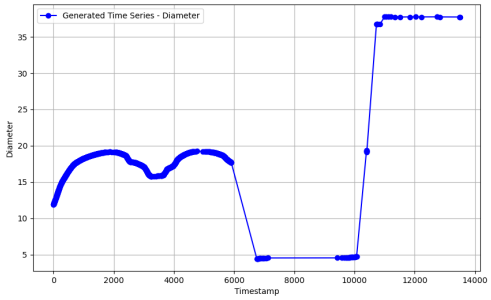

Fig. 2. Keyence measurements for valve lifter.

The continuous data represents a challenge in automating quality control, as the discrete values from the engineering drawings have to be compared with a series of measurements from the Keyence machine. The challenge is to find which exact segments from the time sequence have to be compared to the specifications, which is not a trivial problem. In literature, approaches exist that use statistics, e.g., Extreme Point Selection (EPS), to determine which parts of continuous data should be used for quality control [7]. However, this problem can be avoided using decision mining algorithms as the algorithm automatically determines the significant parts of the measurements if the time sequence is discretized and split into segments. Therefore, setting the Keyence measurements into relation to the requirements and tolerances specified in the ED can enable the mining of a meaningful decision rule for DP1 and thereby support quality control by enabling tracking of which decision logic is actually used to make quality decisions as well as provide a basis for automated quality control.

A decision rule can consist of multiple conditions, which are usually of the form $v(ariable)$ $op(erator)$ $c(onstant)$, for example "measurement1 $>$ 18.5", which are concatenated to form a decision rule. However, in scenarios like the one described above, including the tolerance values in the conditions to embed the measurements in the context of the required dimensions can provide benefits. The corresponding condition is of the form $v(ariable)$ $op(erator)$ $v(ariable)$, for example "measurement1 $>$ tolerance1". These relational conditions capture the relationships between two variables. An example of a relational condition in a loan application scenario is: "IF $amount < amount\_treshold$", where $amount\_threshold$ is referring to contextual data, e.g., compliance regulations, instead of the classic decision rule: "IF $amount < 100.000$". Similarly, an example from the healthcare domain could be: "IF $heart\_rate > max\_threshold$ OR $blood\_pressure < min\_threshold$" In these examples, the relational conditions compare a variable to another variable that is derived from contextual data, such

as regulatory documents or guidelines. This allows the rules to be more flexible and adaptable to changing circumstances. In addition, it provides context information to the user as it is not a constant value but specifies what it relates to; thereby, potential deviations from the intended process can be detected more easily. In the case study, a potential insight could be whether the workpiece quality is assessed based on the required dimensions or on arbitrary values. In addition, constant values might not be exactly the same as the specifications set in the drawing due to learning of the algorithm; e.g., 2.00 was approximated as a threshold instead of the true maximum value of 1.98, which could potentially sum up to account for bigger errors. Therefore, using relational conditions enables more transparent and informative decision rules. An exemplary decision rule for DP1 using relational conditions can be seen in Fig. 1.

Applying decision mining to support quality control in the case study involves several challenges. Firstly, the dimensioning information must be extracted from the engineering drawing in a form that allows for further automated processing. Secondly, the measurements are in the form of time sequence data, which has to be integrated with the dimensioning information in a meaningful way to classify the workpieces accurately. Thirdly, the classification rules have to be communicated to the domain experts transparently [25], i.e., the user has to know according to which rules the workpieces are classified to evaluate if the rules relate to the actual specifications or if unwanted deviations occurred in the quality control process. This process and the related challenges are similar for various workpieces produced using a turning machine, i.e., cylindrical workpieces.

Previous work [21] shows how dimensioning information can be extracted from technical drawings. However, how this information can be implemented as part of the process was not further investigated. Decision mining approaches found in the literature can extract decision rules from event log data [15]. Still, so far, these approaches are not able to relate time sequence data to other data, i.e. embedding the measurements in the context of the required dimensions.

Therefore, integrating information from EDs with time sequence data for decision mining in a meaningful way to automate and optimize the quality assurance process remains to be done. Thus, the primary object of this paper is to bridge the gap between information extraction from EDs and its practical application in decision mining to contribute to seamless and effective automated quality control. This integration is achieved by the DigiEMine approach, which enables the classification of workpieces according to their quality and the extraction of decision rules set in the specifications' context.

Methodology: This paper follows a case study methodology [19]. A case study approach was chosen due to its suitability for in-depth exploration of the real-world complexities involved in implementing and testing the system within this unique use case. The case study and the research questions are introduced as part of this section. Data collection involves implementing and applying the DigiEMine approach on three datasets from the presented scenario. Results are analyzed with regard to their performance as well as their ability to include

context information, allowing us to understand the strengths, challenges, and overall effectiveness of the implementation for this use case.

3 The DigiEMine Approach

The DigiEMine approach is defined through Alg. 1 and consists of three phases. An overview of the approach can be seen in Fig. 3. The gray highlighted lines mark lines that use existing algorithms. As input, the engineering drawing as well as the event log of the production process, are needed.

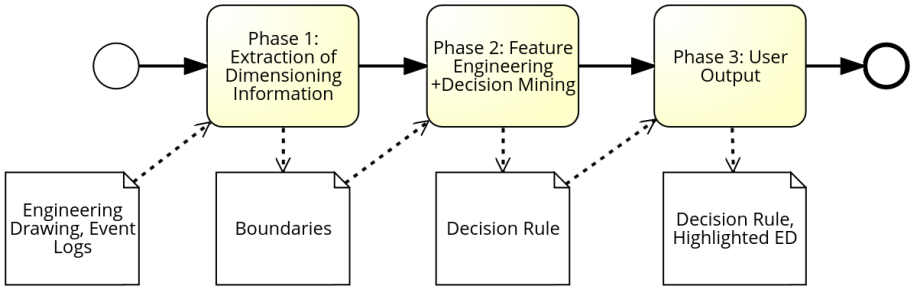

Fig. 3. Overview of the DigiEMine approach (modeled using Signavio©).

In the first phase, dimensioning information, including nominal values and tolerances, is extracted from an engineering drawing using the approach presented in [21]. The nominal values are combined with the tolerances to get upper and lower boundaries, which are used for feature engineering in the next phase.

In the second phase, time sequence data, in this case, the measurement data of the individual workpieces, is extracted from the event logs. New features representing the underlying pattern are extracted from the measurements. These features are combined with the information extracted in Phase 1 to form relational features, which enable the mining of relational conditions. Subsequently, a decision mining algorithm uses the generated features to mine decision rules.

In the third phase, the mined rules are displayed to the user textually. In addition, all tolerances that are part of the resulting decision rule are highlighted in the ED. Therefore, the algorithm's output consists of the decision rule and the highlighted drawing.

Phase 1 Extraction of Dimensions - Alg. 1 Lines 1-3

The algorithm starts by calling a function provided by [21], using an ED as input, returning dimensioning requirements (D) and coordinates of the bounding box (C) of those requirements on the drawing. The requirements are in JSON format, where the nominal value and the upper and lower tolerances are given, e.g. 4.8, +0.2, -0.2. The next step includes using regular expressions to get from the requirements in the above-described form to requirements of the form "lower acceptable value" and "upper acceptable value", e.g., 4.6 and 5 for the example

given above. These values are referred to as boundaries. A data frame (DF)[6] is created, and the lower and upper boundaries (B) are saved as features that stay constant for all instances. The algorithm can be adapted to extract the information from the ED only once and reuse this information every time a new batch of workpieces is produced.

Phase 2 Feature Engineering + Decision Mining - Alg. 1 Lines 4-16 The next step is to get the measurements ($TSvalues$) and status information ($Status$) for all workpieces (W) from event log files. For the investigated use cases, the log files are in yaml format. For each instance, the measurement values of the Keyence measuring process and the result of the MicroVu measuring process, i.e., the status that indicates whether the workpieces are "ok" or "not ok", are extracted from the event log and stored in the data frame. As the decision mining technique used in this approach is a decision tree, a supervised learning technique, a ground truth must be available, here in the form of MicroVu status. The measurement values, i.e., the time sequence values extracted from the log file and the status, are then stored in the data frame for each instance.

Next, the time sequence values are used to create new features(TSF) that reflect the characteristics of the time sequence by applying the feature engineering part of the $EDT\text{-}TS$ approach [23]. $EDT\text{-}TS$ is an approach to discover decision rules that depend on time series data and works by applying different feature engineering methods reflecting different time sequence characteristics. Three types of features are produced: 1) global features that summarize the entire time series, 2) interval-based features that calculate features for subsequences of the time series, and 3) pattern-based features that look at the distribution of values in a time series, e.g., a value has to appear more than five times. The algorithm works by pre-processing event log data to detect time sequence values. Subsequently, different time sequence features are calculated for each instance. Examples of global features generated by EDT-TS are the maximum value, e.g., $diameter\_maximum$, the slope of a time series, or more complex values such as a Fourier transform. The time series is divided into intervals for interval-based features, and features are calculated for each interval. The time series can be split by measurement points or time spans. Examples include the mean, maximum, and percentage change of each interval, e.g., $diameter\_segment2\_percentchange$, referring to the percent change of values in the second interval. Per default the time series is split into three, five and ten intervals, however this can be manually adapted to fit the specific use case. In this case, the default intervals were used. Pattern-based features consider the distribution of values in the time series. The algorithm identifies values that occur more often in one class than in another. These values are then used as thresholds to create binary features. For example, if the value 26 occurs more than four times in the temperature time series, the feature $temperature\_list.count(26.0) >= 4.0$ would be set to $True$. As all potential features are calculated for each instance, the number of features increases exponentially, leading to increased computational complexity. To avoid work-

[6] Using pandas, https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html

Algorithm 1 DigiEMine Approach

Input: ED, Workpiece Event Logs WL
Output: Textual Decision Rules, Highlighted ED

$D, C \leftarrow$ dimensions and coordinates from ED ▷ Using [21]
$DF \leftarrow$ new data frame
$DF[B] \leftarrow$ upper and lower boundaries from D using regex
for $W \in WL$ **do**
    $DF[W][TSvalues] \leftarrow$ measurement values for W
    $DF[W][Status] \leftarrow$ final status of W
end for
$TSF \leftarrow$ generate time sequence features ▷ Using [23]
for $tsf \in TSF$ **do**
    if $abs(correlation(tsf, Status)) > 0.1$ **then**
        $DF[RelevantTSF] \leftarrow tsf$
    end if
end for
$DF[RelationalF] \leftarrow$ generate relational features(DF, B) ▷ Using [22]
$DM \leftarrow$ build decision tree using DF
$DR \leftarrow$ generate decision rules using DM
for $Condition\ C \in DR$ **do**
    $RelevantBoundary \leftarrow$ use regex to find which boundary included in C
    if $RelevantBoundary = \emptyset$ **then**
        for $b_{lower}, b_{upper} \in B$ **do**
            if $b_{lower} < C_{value} < b_{upper}$ **then**
                $RelevantBoundary \leftarrow B$
            end if
        end for
    end if
    if $RelevantBoundary = \emptyset$ **then**
        for $b \in B$ **do**
            if $min_{difference}(C_{value}, b)$ **then**
                $RelevantBoundary \leftarrow B$
            end if
        end for
        output warning to user
    end if
    $CB \leftarrow$ coordinates for $RelevantBoundary$ using C
    draw rectangle around CB on ED
end for
return textual decision rules and highlighted ED

ing with potentially irrelevant features, the correlation between the engineered features and the resulting outcome, i.e., the status (OK/NOK), is computed. This is done using the Phi Coefficient for two outcome classes and the Pearson correlation coefficient for more than two outcome classes. Only features with an absolute correlation coefficient of at least 0.1 are considered potentially relevant ($RelevantF$) and used for the next steps. The threshold of 0.1 worked well for the tested cases but can be adapted.

In the next step, relational features are created to enable extracting relational conditions instead of conditions using constant values. The relevant features ($RelevantF$) are combined with the boundaries B to create relational features ($RelationalF$), i.e., features of the form $measurement1 <= boundary1$, to set the measurements in relation to the boundaries. After all potential combinations of measurements and boundaries are created, they are calculated (true/false) for each instance, leading to features such as "$measurement1 <= boundary1 ==$ TRUE". The last step in phase 2 consists of mining decision rules. Decision rule mining is usually done using classification algorithms; see [15]. Decision trees are especially useful as these produce white-box decision models and allow for the generation of textual decision rules. Therefore, decision trees are also used in this use case[7] The created features ($RelationalF$ and $RelevantF$) are used as input to the decision tree implementation. Decision trees recursively split the feature space into distinct regions based on the values of input features. Each internal node within the tree represents a decision condition based on a specific feature, with different branches from nodes corresponding to different possible feature values [4]. This partitioning process continues until a stopping criterion is reached, typically when the data points within the leaf node are predominantly of a single class. The resulting decision model (DM) contains a tree structure, enabling the classification of new instances by traversing the tree from the root node to a leaf node based on the feature values of the instance. As a result, textual decision rules containing one or more conditions are generated(DR).

Phase 3 User Output - Alg. 1 Lines 17-37

In the last phase, the mined decision rule has to be communicated to the domain expert. In production, not all specifications in the ED are relevant to the result. The further use of the workpiece is often decisive in determining which features are essential and which are less critical. However, the production process can also influence which dimensions are most critical, e.g., chips might form on one specific part of a workpiece. Therefore, knowing which parts of the workpiece are most relevant to the outcome enables additional insight. To enable a visual understanding of which dimensions contribute to the classification of an instance, all boundaries that are part of the decision rule are mapped to the corresponding dimension and highlighted in the original ED. Therefore, regular expressions are used to extract the boundaries ($RelevantBoundary$) used for each condition (C). The condition's specific value (C_{value}) is analyzed if the conditions do not contain relational features. The value is mapped to a dimension if it lies between the upper and lower boundary (b_{lower}, b_{upper}). If neither approach can map conditions to dimensions, a textual warning is given to the user as dimensions and rules do not overlap, and conformance issues could be involved. In addition, the nearest boundary for each value, b_{lower} or b_{upper}, is analyzed, with "near" being defined as the minimum absolute difference, as this might be the appropriate boundary. This mapping is speculative, and therefore, the warning is displayed. Lastly, for all dimensions that are part of a condition, the coordi-

[7] Here, the Scikit-learn implementation of CART is used, see [18]

nates of the bounding boxes (CB)are retrieved to highlight the dimensions on the ED, which is shown to the user and stored.

4 Case Study Findings

Algorithm 1 was implemented using Python and tested on three datasets stemming from the production of cylindrical workpieces. The implementation is available online[8], including all used datasets and the full results.

As this is, to the best of our knowledge, the first approach that enables integration of time sequence data and relational features, the evaluation focuses on feasibility and applicability. The feasibility of the approach was shown by implementing it. The approach is tested on three datasets to evaluate its applicability. The resulting conditions are compared to EDT-TS results, which allows for the integration of time sequence data but not the generation of relational features. Accuracy is calculated for EDT-TS and DigiEMine to analyze if the application of DigiEMine leads to changes in performance.

The datasets used for the case study should stem from a cylindrical workpiece production process. In addition, the engineering drawing of the workpiece or at least the tolerance values should be available. Furthermore, the dataset should contain measurements of the workpieces and information about whether the workpieces are "ok" or "not ok", i.e., some ground truth has to be known to learn the decision tree as well as to evaluate performance of the mined decision. As it is challenging to find appropriate datasets, we used three datasets based on the case study described in Sect.2: one real-life dataset ("Valve Lifter") corresponding to the case presented in Sect. 2, a second dataset taken from the same scenario but involving a different workpiece, called "Turm" and a third, synthetically created, dataset. The third dataset ("Synthetic") is similar to the "Turm" dataset but includes generated time sequence values.

Results: Tab. 1 shows the accuracy values achieved by the DigiEMine approach and *EDT-TS* approach for the three datasets and an excerpt from the mined decision rules.

Table 1. Evaluation results for the datasets Valve Lifter, Turm and Synthetic.

DataSet / Approach		DigiEMine		EDT-TS	
	N	Accuracy	Example Condition	Accuracy	Example Condition
Valve Lifter	37	1	boundary10>=segment8_min is TRUE	0.75	segment5_min <= 15.79
Turm	33	0.75	boundary2>segment6_max is FALSE	0.75	segment1_max <= 77.73
Synthetic	70	1	boundary4<segment8_max is TRUE	0.86	segment5_max <= 22.17

The table shows that the accuracy values are medium to high for all datasets and approaches. However, the values achieved by DigiEMine are at least as high and, in some instances, even considerably higher than the results achieved by *EDT-TS*. The conditions look similar in each approach. The *EDT-TS* conditions involve minimum or maximum segment values which are set in comparison to a

[8] https://github.com/bscheibel/digiemine

threshold value instead of a boundary. In the case of the valve lifter, EDT-TS extracted dimensions that are not precisely accurate, i.e., the exact specification would be 15.2 as the maximum value compared to the extracted value of 15.79; similarly, for the synthetic dataset, the specified maximum value is 22.1, whereas 22.17 was mined as maximum in EDT-TS. These are minor differences but can accumulate and may account for the differences in accuracy. In addition, even minor differences might be impactful in production when exact measurements are needed for specific workpieces. For the "Turm" dataset, the decision

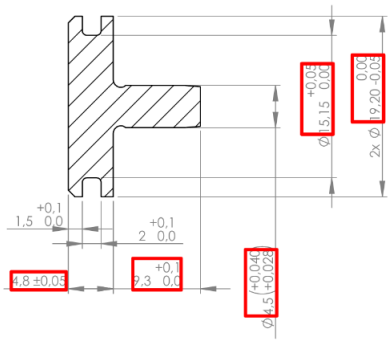

Fig. 4. Valve lifter engineering drawing with highlighted dimensions.

rule discovered by EDT-TS involves only one condition. However, this condition does not include a value related to the dimensions but is an artifact created by measuring. Therefore, despite high-performance values, the discovered conditions include no valid classification rule. The DigiEMine conditions involve comparing a segment's minimum or maximum to a boundary and specifying if those conditions should be $True$ or $False$. The full decision rules contain concatenations of three or more of those conditions, each comparing segments and boundaries. Therefore, users working with these decision rules can assess which segments have to be compared to which specifications set by the engineering drawing. In addition, Fig. 4 shows the visual output for the described use case: the dimensions used as part of the decision rule are highlighted in the original ED.

5 Discussion

The application of the DigiEMine approach in the case study shows that the approach is feasible and able to discover decision rules, including time sequence and contextual data, and to set them in relation to each other, thereby providing an answer to the **RQ** stated in Sect. 1. The results indicate that the performance is at least as high or even higher than without the inclusion of dimensioning requirements. The high accuracy values for DigiEMine can be due to the ability

to use the exact specifications instead of having to approximate them using the available instances. This is also a benefit for classifying new instances, as these might include values not seen during testing.

In the results, excerpts of the mined decision rules are given, showing that the measurements are always set in relationship to a boundary, i.e., a dimension from the ED. A visual output, in the form of an engineering drawing with highlighted dimensions, is also provided. This can further support the employees supervising the process as well as improve the understanding of which dimensions are decisive for the outcome of the process. If measurements cannot be linked to the dimensions found in the ED, this could indicate that the measuring process cannot detect quality-relevant attributes; for example, only the diameter is measured, which is irrelevant to the process outcome as only surface attributes like flatness are decisive for the result. Alternatively, it could also indicate that the classification of workpieces is not based on the requirements specified in the ED. If this is the case, a potential conformance issue could be involved. On the other hand, not all dimensions found in the ED can be linked to measurements, as not all requirements can be measured using one measuring machine.

The integration of dimensions and tolerances through the use of an algorithm can also lead to the introduction of errors. Therefore, ideally, the dimensions are integrated into the CAD model and can be read automatically. However, as mentioned in Sect.2 this is not industry standard. Extracting the dimensions manually from a file is labor-intensive and error-prone, as an engineering drawing can include hundreds of dimensions for complex workpieces. Therefore, the automatic extraction can be used as a starting point and combined with a manual check to ensure the extracted dimensions are valid. This application might be particularly interesting for runtime application, as changes in the contractual basis can be detected. If the measured dimensions do not change accordingly, compliance issues can be registered, and employees notified accordingly.

The approach can be used in addition to manual checking or for fully automated pre-checking, thereby providing a smoother and more efficient manufacturing process, reducing quality assurance time, and providing the best quality workpieces to customers. The application of the DigiEMine approach can provide benefits to manufacturing companies that are on their way to digitalization but still use some form of legacy data. Specifically companies that produce smaller batches can benefit from this approach, as smaller batches mean less training data. Relating the measurements to the boundaries might lead to faster learning with less training data, as the boundaries do not have to be estimated using many instances but can be learned from the information in the provided engineering drawing. If this approach leads to faster learning will be evaluated in future work. In addition, this approach also allows for automated updating if customer requirements, i.e., the engineering drawing, change.

The approach can be generalized to a scenario where time sequence data (e.g., sensor data) and additional context data (e.g. data extracted from regulatory documents or guidelines) should be combined to form meaningful decision rules, which is conceivable in many scenarios, such as healthcare or logistics.

An example in healthcare would be monitoring cardiac conditions where blood pressure or heart rate measurements are compared to clinical guidelines, e.g., the European Society of Cardiology (ESC) guidelines to diagnose specific illnesses. An example from the logistics domain could be the combination of customer requirements regarding transport conditions (e.g., temperature) extracted from emails with the temperature measurement values. This paper provides an approach for a specific use case from manufacturing. However, the fundamental approach is similar, regardless of which data should be integrated. It consists of the following steps: First, a **Data Collection** step, where contextual data and process data are gathered. After that, **Feature Engineering** has to be performed, where features are engineered from unstructured data, and subsequently, relational features are generated. If the decision points are already known, **Decision Mining** can be performed in the next step. Depending on the use case, different decision mining algorithms can be applied. If the decision points are not known, process mining and decision point discovery have to be performed beforehand. Quality metrics, see for example [25] and user feedback can be used to assess and validate the resulting decision rules and potentially initiate a remining of the decision rules, leading to an iterative process. Lastly, in the **Output** step, textual decision rules are displayed to the user; in addition, visualizations can be generated to help the user gain insights into the process.

Limitations and threats to validity: The most significant limitation is the generalizability, as the implementation and evaluation of the approach are set in the context of the case study. In addition, only silhouette measuring was used; thereby, not all quality-relevant criteria can be evaluated. More testing must be done to assess the generalizability of this approach to other kinds of workpieces and in other settings. Furthermore, we currently assume that each dimension is unique. If multiple dimensions with the same values exist, we cannot accurately map the conditions to the dimensions in the drawing. As the classification technique is a supervised learning technique, a ground truth is necessary to learn the decision rules, either by having a second measurement as in the proposed scenario or by including manual measurements. As mentioned above, an analogous usage of this approach in scenarios with time sequence data and additional contextual information is conceivable. However, the approach must be adapted and tested in other scenarios in future work.

6 Related Work

An extensive research domain investigates the **extraction of information** from EDs in CAD formats (DXF, DWG, STEP or IGES) [28,30,31] or from scanned images using vectorization and OCR [17]. DigiEMine uses the approach described in [21] as this is the only approach extracting information from PDF format, which brings together the ability to obtain textual information and to be more accurate than by using OCR. Other kinds of contextual information are investigated, such as extracting information from regulatory documents[6,27] or using news sentiment analysis for additional context [29].

Time series and time sequence data as context data, e.g., additional sensor data or constraints, has been used in multiple scenarios to improve process mining or process monitoring techniques [10,24]. Similarly, existing work integrates time sequence data in process and decision mining by using feature engineering methods [2,9,23]. However, these approaches did not analyze how time sequence data can be connected with additional context data.
Decision Mining includes algorithms for mining decision points from processes and classification techniques to mine the corresponding decision rules. A variety of decision mining approaches exist, focusing on different aspects, such as finding overlapping rules [16], aligning control and data flow to discover decision rules [13], integrating time sequence data [9,23] or mining rules that involve relationships between features, i.e. relational decision rules, [3,14,22]. An overview can be found in [15]. This work integrates existing decision mining approaches to work with time sequence-based and relational features.
A multitude of works investigate data mining for **quality control in manufacturing** [8,11,26]. These overviews include scenarios where techniques classify instances according to their outcome. Often used techniques include neural networks, support vector machines, k-means, and decision trees. Decision trees are specifically used to generate flowcharts to classify outcomes based on different features. Some works use time sequence data and different discretization methods to generate higher-level features.

To the best of our knowledge, DigiEMine is the first approach that combines the information contained in EDs with time sequence measurements, thereby bridging multiple fields. DigiEMine is flexible in that the techniques used can be replaced by other appropriate techniques, e.g., the generation of time sequence features can also be done using other discretization methods.

7 Conclusion

The case study presented in this paper shows how the DigiMine approach can support quality control processes in manufacturing. DigiEMine enables the integration of context information, specifically dimensioning information from EDs with time sequence data, i.e., workpiece measurements, to enable the mining of relational decision rules, providing more transparency in quality control processes. The evaluation showed that the approach is feasible and produces results setting time sequence data in relation to context data, achieving accuracy values between 0.75 and 1 for the tested datasets. Further testing and generalization of DigiEMine for different scenarios is planned for future work. Moreover, we aim to use the approach in runtime decision mining scenarios to test the hypothesis that this approach allows for faster mining of accurate decision rules. Furthermore, a user study can evaluate different presentations of the textual rules (e.g., in the form of trees or tables) as well as the visualizations in the drawing.
Acknowledgments: This work has been partly supported and funded by the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 881843.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action. Springer Berlin Heidelberg, 2 edn. (2016). https://doi.org/10.1007/978-3-662-49851-4_1
2. Banham, A., Leemans, S.J.J., Wynn, M.T., Andrews, R., Laupland, K.B., Shinners, L.: xPM: Enhancing exogenous data visibility. Artificial Intelligence in Medicine p. 102409 (Sep 2022). https://doi.org/10.1016/j.artmed.2022.102409
3. Bazhenova, E., Buelow, S., Weske, M.: Discovering Decision Models from Event Logs. In: Business Information Systems. vol. 255, pp. 237–251. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-39426-8_19
4. Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman and Hall/CRC, New York (Oct 2017). https://doi.org/10.1201/9781315139470
5. Brunk, J.: Structuring Business Process Context Information for Process Monitoring and Prediction. In: Conference on Business Informatics. vol. 1, pp. 39–48 (Jun 2020). https://doi.org/10.1109/CBI49978.2020.00012
6. Chen, Q., Winter, K., Rinderle-Ma, S.: Predicting Unseen Process Behavior Based on Context Information from Compliance Constraints. In: Business Process Management Forum. pp. 127–144. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-41623-1_8
7. Colosimo, B.M., Moya, E.G., Moroni, G., Petrò, S.: Statistical Sampling Strategies for Geometric Tolerance Inspection by CMM. Economic Quality Control **23**(1) (Jan 2008). https://doi.org/10.1515/EQC.2008.109
8. Dogan, A., Birant, D.: Machine learning and data mining in manufacturing. Expert Systems with Applications **166**, 114060 (Mar 2021). https://doi.org/10.1016/j.eswa.2020.114060
9. Dunkl, R., Rinderle-Ma, S., Grossmann, W., Fröschl, K.A.: A Method for Analyzing Time Series Data in Process Mining: Application and Extension of Decision Point Analysis. In: Information Systems Engineering in Complex Environments. pp. 68–84. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-19270-3_5
10. Ehrendorfer, M., Mangler, J., Rinderle-Ma, S.: Assessing the Impact of Context Data on Process Outcomes During Runtime. In: Service-Oriented Computing. pp. 3–18. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_1
11. Koeksal, G., Batmaz, I., Testik, M.C.: A review of data mining applications for quality improvement in manufacturing industry. Expert Systems with Applications **38**(10), 13448–13467 (Sep 2011). https://doi.org/10.1016/j.eswa.2011.04.063
12. Labisch, S., Weber, C.: Technisches Zeichnen Selbstständig lernen und effektiv üben. Viewegs Fachbücher der Technik, 3 edn. (2008)
13. de Leoni, M., van der Aalst, W.M.P.: Data-aware process mining: discovering decisions in processes using alignments. In: Applied Computing. p. 1454. ACM Press, Coimbra, Portugal (2013). https://doi.org/10.1145/2480362.2480633
14. de Leoni, M., Dumas, M., García-Bañuelos, L.: Discovering Branching Conditions from Business Process Execution Logs. In: Fundamental Approaches to Software Engineering. pp. 114–129. Springer, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37057-1_9
15. de Leoni, M., Mannhardt, F.: Decision Discovery in Business Processes. In: Encyclopedia of Big Data Technologies, pp. 1–12. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-63962-8_96-1

16. Mannhardt, F., Leoni, M.d., Reijers, H.A., van der Aalst, W.M.P.: Decision Mining Revisited - Discovering Overlapping Rules. In: Advanced Information Systems Engineering. pp. 377–392. Springer, Cham (Jun 2016). https://doi.org/10.1007/978-3-319-39696-5_23
17. Moreno-García, C.F., Elyan, E., Jayne, C.: New trends on digitisation of complex engineering drawings. Neural Computing and Applications **1**, 1–18 (Jun 2018). https://doi.org/10.1007/s00521-018-3583-1
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D.: Scikit-learn: Machine Learning in Python. The Journal of Machine Learning Research **12**, 1–6 (2011)
19. Recker, J.: Scientific Research in Information Systems. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-30048-6
20. Reinkemeyer, L. (ed.): Process mining in action: principles, use cases and outlook. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-40172-6
21. Scheibel, B., Mangler, J., Rinderle-Ma, S.: Extraction of dimension requirements from engineering drawings for supporting quality control in production processes. Computers in Industry **129**, 103442 (Aug 2021). https://doi.org/10.1016/j.compind.2021.103442
22. Scheibel, B., Rinderle-Ma, S.: Comparing decision mining approaches with regard to the meaningfulness of their results. arXiv:2109.07335 [cs] (Sep 2021)
23. Scheibel, B., Rinderle-Ma, S.: Decision Mining with Time Series Data Based on Automatic Feature Generation. In: Advanced Information Systems Engineering. vol. 13295, pp. 3–18. Springer (2022). https://doi.org/10.1007/978-3-031-07472-1_1
24. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: Business Process Management, vol. 12168, pp. 202–219. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-58666-9_12
25. Wais, B., Rinderle-Ma, S.: Towards a Comprehensive Evaluation of Decision Rules and Decision Mining Algorithms Beyond Accuracy. In: Advanced Information Systems Engineering. pp. 403–419. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-61057-8_24
26. Wang, K.: Applying data mining to manufacturing: the nature and implications. Journal of Intelligent Manufacturing **18**(4), 487–495 (Aug 2007). https://doi.org/https://doi.org/10.1007/s10845-007-0053-5
27. Winter, K., Rinderle-Ma, S.: Detecting Constraints and Their Relations from Regulatory Documents Using NLP Techniques. In: On the Move to Meaningful Internet Systems. pp. 261–278. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_15
28. Ye, B., Liu, J., Wu, B., Wu, C.: New method of feature recognition from engineering drawings based on multi-granularity information acquisition. International Conference on Fuzzy Systems and Knowledge Discovery **5**, 129–133 (2009). https://doi.org/10.1109/FSKD.2009.802
29. Yeshchenko, A., Durier, F., Revoredo, K., Mendling, J., Santoro, F.: Context-Aware Predictive Process Monitoring: The Impact of News Sentiment. In: On the Move to Meaningful Internet Systems. pp. 586–603. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-030-02610-3_33
30. Zhang, H., Li, X.: Data Extraction from DXF File and Visual Display. In: HCI International 2014. vol. 434, pp. 286–291 (2014). https://doi.org/10.1007/978-3-319-07857-1_51

31. Zhang, J., Zhao, L., Hao, Y.: Multi-level block information extraction in engineering drawings based on depth-first algorithm. Advanced Materials Research **468-471**, 2100–2103 (2012). https://doi.org/10.4028/www.scientific.net/AMR.468-471.2100