applied sciences

MDPI

Article

Precision, Fitness, Generalization, and Simplicity as Quality Dimensions for Decision Discovery Algorithms

Sam Leewis *, Koen Smit and Annemae van de Hoef

Research Group Digital Ethics, HU University of Applied Sciences Utrecht, 3584 CH Utrecht, The Netherlands; koen.smit@hu.nl (K.S.); annemae.vandehoef@hu.nl (A.v.d.H.)
* Correspondence: sam.leewis@hu.nl

Featured Application

The proposed precision, fitness, generalization, and simplicity metrics for decision mining can be directly applied in domains where transparency and accountability of decision-making are important. For example, in public administration, decision discovery algorithms can be evaluated and improved when applied to regulatory contexts. By assessing whether a discovered decision model is both precise and fit, organizations can ensure that automated or semi-automated decision-making processes remain aligned with legal requirements and public values. Beyond regulatory compliance, the metrics provide a practical tool for practitioners to identify over- and under-specifications in decision logic, enabling an iterative refinement of decision models. This facilitates not only more reliable operational decisions but also contributes to broader societal goals such as fairness and trust in digital decision-making systems.

Abstract

Operational decisions significantly influence organizational performance and individual well-being. Decision mining offers a method to discover and analyze decision logic from decision logs, enhancing decision-making processes. However, evaluating the quality of decision discovery algorithms remains a challenge. While precision, fitness, generalization, and simplicity are well-established quality dimensions in process mining, their adaptation to the decision mining domain is underexplored. This study adapts these four dimensions to the necessary characteristics of decision models, providing a framework for evaluating decision discovery algorithms. Using a design science research approach, we develop tailored metrics and functions and demonstrate their application through a practical example of environmental permit management modeled in Decision Model and Notation (DMN). Precision measures how the discovered decision model reproduces the observed fact types and values from the decision log, detecting over-specification in the decision model. Fitness evaluates how completely the decision model covers the behavior in the decision log, identifying missing or under-specified elements in the decision model. Generalization assesses the model's robustness to unseen decision cases by quantifying how well the discovered decision model performs beyond the training data. Simplicity captures the complexity in the discovered decision model in relation to a human actor-specified threshold. These insights guide decision model improvements, contributing to higher transparency, accountability, and fairness in operational decision-making processes. This research bridges a gap in the body of knowledge by providing a concrete methodology for evaluating decision discovery algorithms. The results support organizations in aligning decision models with regulatory requirements and public values, while also laying a foundation for future research.

check for updates

Academic Editor: George Drosatos

Received: 4 September 2025
Revised: 9 October 2025
Accepted: 9 October 2025
Published: 15 October 2025

Citation: Leewis, S.; Smit, K.; van de Hoef, A. Precision, Fitness, Generalization, and Simplicity as Quality Dimensions for Decision Discovery Algorithms. *Appl. Sci.* **2025**, *15*, 11060. https://doi.org/10.3390/app152011060

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Keywords: precision; fitness; generalization; simplicity; decision discovery; quality dimensions

1. Introduction

High-volume operational decisions are of great value for organizations and their customers [1]. This is, in one way, due to the large volume of data resulting from these decisions, which could be used for creating and making smarter decisions [2,3]. Furthermore, operational decisions could have a positive or negative impact on a person's day-to-day life. For example, a government's operational decision might involve the issuance of a residence permit, a decision that can have a substantial impact on an individual's overall welfare. Another example of the impact of operational decision-making would be in hospitals where a wrongly executed decision could directly affect a person's quality of life. In this paper, we adhere to the following definition of an operational decision: *"The act of determining an output value from a number of input values, using decision logic defining how the output is determined from the inputs"* [4]. An important component of how the decision is made is referred to as 'decision logic' in the above definition. In this paper, we adhere to the following definition of decision logic: "*a collection of business rules, business decision tables, or executable analytic models to make individual business decisions*" [4]. These definitions identify business rules and decision tables as specific components, providing potential benefits when separating and managing them from the rest of the software and processes.

This evolution has led to the separation of various concerns (SoCs) over time. Separating concerns started with the separation of the 'data concern' from software in the 1970s, followed by the separation of the 'user interface concern' in the 1980s and the 'workflow/process concern' in the 1990s, as noted by [5]. Studies conducted by Boyer and Mili [6] and Smit [7] indicate that a logical progression in the field of IT involves isolating business rules as an independent 'concern.' This line of thinking has become particularly relevant for the development of intelligent systems, where decisions and their logic are increasingly separated from processes and data to support adaptability and transparency. Several researchers go even further with the SoC and separate decisions from data and processes [8–12]. Provided the impact of decisions, its separate and explicit management becomes even more important. However, it could be for several reasons that an organization has no explicit decision management. One of the reasons could be that the organization is familiar with the fact that a certain decision exists, but the choice is made not to model the actual decision. Another reason could be that the organization is not familiar with the existence of certain decisions and is therefore unmanaged. Explicit decision modeling, however, can contribute to the design of decision support systems by enhancing consistency, explainability, and automation of decision logic. For either reason, having an explicit decision model certainly has benefits for the decision management of an organization. For example, one of the industry standards in the realm of decision modeling is the Decision Model and Notation (DMN) from the Object Management Group (OMG) [4]. DMN supports the use of comprehensible diagrams that are designed to be understood by organizational stakeholders. These diagrams serve as a foundation for constructive discussions, leading to consensus regarding the scope and nature of decision-making processes, but also support validation by stakeholders. Furthermore, by breaking down requirements into graphical components, these objectives reduce the complexity and associated risks of automating decision processes. Decision tables, as a part of this standard, enable the structured definition of business rules. In addition, they simplify the development of information systems, offering specifications that can be automatically verified and executed, i.e., using Friendly Enough Expression Language (FEEL) modeling. It is also designed to align, both from a modeling

and technical perspective, with the Business Process Management and Notation (BPMN) standard [13]. Ultimately, it facilitates the creation of a library of reusable decision-making components, fostering efficiency and consistency in the decision-making process. DMN therefore contributes to the separation of concerns of decisions from data and processes. This makes it particularly suitable for embedding in decision support systems that depend on rule-based transparency and decision traceability [14,15].

Combining (business) data analytics with decision management solutions is something that is conducted often [1]. Recorded data resulting from previously made decisions can be used for creating and making smarter future decisions [2,3]. The added value of collecting the relevant data related to these decisions and, in turn, analyzing the data resulting from these decisions could be of added value for designing, implementing, and executing said decisions [16]. Therefore, discovering and analyzing decisions and its related decision logic is relevant for increasing the quality of operational decision-making [17]. A method that could be utilized in order to discover decisions and its related decision logic is decision mining [10–12]. Decision mining is "*the method of extracting and analyzing decision logs with the aim to extract information from such decision logs for the creation of business rules, to check compliance to business rules and regulations, and to present performance information*" [12]. Decision mining has the potential to positively ground public values such as (1) transparency (by making the decision-making explicit) [16,17], (2) accountability (decision-making by whom and based on which data) [18–21], and (3) fairness (revealing biases or inconsistencies in the decision-making could lead to more equitable outcomes) [20,22]. As such, decision mining forms a promising basis for data-driven decision support systems that aim to operationalize responsible decision-making.

Due to this reason, the algorithms used in decision mining should be evaluated themselves in order to have a positive effect on public values. Similar to how the FAIR principles [23] have established findability, accessibility, interoperability, and reusability as guiding criteria for the responsible management of data, evaluation frameworks for decision discovery algorithms should ensure comparable qualities such as transparency, traceability, and reusability of decision logic.

The evaluation of decision discovery algorithms could be conducted through the utilization of quality dimensions. Quality dimensions ensure that discovery algorithms can be characterized, evaluated, and selected for a certain purpose. Subsequently, the discovered process or decision model (the outcome of the discovery algorithms) could be held in some regard with the characterization of the discovery algorithms. However, on the one hand, to the knowledge of the authors, the current body of knowledge on decision mining does not contain detailed contributions on quality dimensions. On the other hand, we know that process mining is a neighboring field of study to decision mining that shows a high similarity [12]. Process mining is defined as "*the discovery, monitoring and improvement of real processes by extracting knowledge from event logs readily available in today's information systems*" [24]. Three types of process mining exist, which are as follows [24]: (1) process discovery, (2) conformance checking, and (3) process improvement. The high similarity between decision mining and process mining can guide us towards exploring the quality dimensions of Precision, Fitness, Simplicity, and Generalization in the process mining domain for the utilization in the decision mining domain [25–27]. However, while these quality dimensions are well developed for processes, adapting them to decisions involves new challenges due to differences in structure, granularity, and logic representation. Additionally, while optimization approaches in generalized multi-scale decision systems highlight the impact of representational choices on model quality [28,29], decision discovery operates under stricter constraints: decision logic-based models are often a translation from policy or legislation and changing this is highly undesirable as it

risks immediate regulatory non-compliance. This further underlines the need for quality dimensions such as precision, fitness, generalization, and simplicity to evaluate the fidelity of discovery algorithms.

Encouraged by the exaptation strategy from Hevner and Gregor [30], we leverage quality dimensions of process mining for the field of decision mining. Therefore, we aim to answer the following research question in this study: *How can the precision, fitness, generalization, and simplicity quality dimensions from process mining be adapted to effectively evaluate decision discovery algorithms in the decision mining domain?*

This contribution fills an identified gap by tailoring evaluation metrics for decision mining and showing their practical application. While existing studies in process mining provide well-defined quality dimensions such as precision and fitness, their adaptation to the decision mining domain remains underexplored. This lack of a concrete methodology for evaluating decision discovery algorithms impedes the effective implementation of decision management solutions in practice [31,32]. This research aims to bridge this gap by proposing precision, fitness, generalization, and simplicity metrics tailored to decision mining and demonstrating their application through a practical example.

The remainder of the paper is structured as follows: Firstly, the background and related work are discussed containing concepts relating to conformance checking in process mining and decision mining, and the process and decision discovery algorithm quality dimensions. Secondly, the implementation of the decision quality dimensions of precision, fitness, generalization, and simplicity is discussed using a running example. Finally, the study and its outcomes are discussed, and conclusions are drawn to answer the research question, which is followed by possible future research directions.

2. Background and Related Work

Several application domains of intelligent systems, such as healthcare diagnostics, environmental regulation, loan approval, and supply chain optimization, rely heavily on structured decision logic. Yet, the accuracy and interpretability of this logic remain underevaluated in most operational contexts. Recent work has highlighted the growing need for explainable and auditable decision support systems (e.g., [33]). While many decision support systems embed decision rules, few employ a systematic quality evaluation of the decision models they contain. This study addresses this gap by proposing precision, fitness, generalization, and simplicity metrics tailored to decision mining, enabling decision support systems to improve the quality of their internal decision logic. This is particularly relevant for applications where decisions have legal, financial, or medical consequences and require both correctness and transparency.

Several methods exist that focus on discovering, checking, and improving patterns from data. The field of data mining focuses on knowledge discovery, or more specifically, aggregation patterns [34], where the specific fields of process mining [24] and decision mining [12] focus on process and decision patterns. Process mining and decision mining are two fields that have a similar approach. This approach could be summarized into three phases: discovery of processes or decisions, checking processes or decisions on conformance, and/or enhancing or improving processes or decisions. Previous studies exist that focus on discovering decisions from data [8,9,11,35]. Mannhardt et al. [36] introduced methods to discover overlapping rules in decision mining, while Scheibel and Rinderle-Ma [37] proposed time series-based decision mining techniques with automatic feature generation. Furthermore, Banham et al. [31] and Wais and Rinderle-Ma [32] emphasized the need for a comprehensive evaluation framework for decision rules, moving beyond simple accuracy metrics to consider conformance and performance dimensions. Nonetheless, research exists on multi-scale decision systems, which states that that the selection of attributes

scales has a direct effect on classification performance and rule extraction [28,29]. However, such optimization presupposes that the underlying decision logic can be freely adjusted, which is highly undesirable as it risks immediate regulatory non-compliance. Therefore, a major gap in the literature is the lack of a concrete methodology for evaluating decision discovery algorithms using quality dimensions such as precision, fitness, generalization, and simplicity.

Precision, fitness, generalization, and simplicity are well-established metrics in data mining and process mining. In data mining, precision is defined as the proportion of true positive predictions among all positive predictions, while fitness, often referred to as recall, measures the proportion of actual positives that are correctly identified [34]. These metrics provide a robust framework for evaluating the accuracy and reliability of classification algorithms. In process mining, the focus lies on diagnosing the event logs and the process models. This is (partly) conducted by analyzing the used discovery algorithm which discovers process models from event logs. This diagnosis includes the characterization of the discovery algorithm using the process quality dimensions [25,26]. Four quality dimensions exist to characterize a discovery algorithm for process mining [24–27,38].

In contrast to prior work, our method specifically focuses on evaluating the structure and behavior of decision logic models using quality dimensions derived from process mining. Where existing approaches typically assess individual rules or trees using predictive performance (e.g., accuracy or F1-score), our approach evaluates decision models holistically with respect to their conformance to actual decision behavior. In particular, we adapt process-oriented concepts such as 'trace fitness' and 'model overgeneralization' to decision artifacts, including fact types, fact values, and layered decision structures such as those in DMN. This allows for more informative assessments in decision support system contexts, where rule transparency, reuse, and auditability are crucial.

However, while precision, fitness, generalization, and simplicity are widely applied in data mining and process mining, their adaptation to decision mining requires careful consideration. Decision mining focuses on discovering decision logic from decision logs rather than classifying discrete outcomes. Therefore, the definitions of precision, fitness, generalization, and simplicity must be recontextualized to evaluate whether a discovered decision model accurately reflects the decision patterns captured in the decision log. The adoption of these quality metrics in decision mining remains limited, and existing studies often overlook how the distinct characteristics of decision models, such as their layered structure (for example in DMN's Decision Requirements Diagram and Decision Logic Layer), influence the evaluation process [4]. Additionally, while precision, fitness, generalization, and simplicity are widely used in classification contexts, their specific adaptation to decision mining requires a more tailored approach that considers decision-specific artifacts like fact types and fact values [12].

Several key concepts are defined to better understand the quality dimensions in relation to process mining and decision mining, starting with the key concepts of quality dimensions in the process mining domain. The first concept is the event log: a log file containing detailed information about the activities that have been executed, as shown on the left in Figure 1. The event log consists of a case ID, events, and related attributes such as activity, time, costs, and resources [24]. The second concept is the process model (such as BPMN [13] or Petri nets [39]): a (graphical) representation of the flow of work of activities, shown on the right in Figure 1. The third concept is behavior: a combination of activities which could be combined into a process, as shown in Figure 1. The fourth and last concept is a trace: a single case under a unique case ID of a specific combination of activities which could be combined into a process. An example of a trace of events would

be: A → B → C (trace for case ID 1) or A → D (trace for case ID 2), as shown in Figure 1 on the left and right.

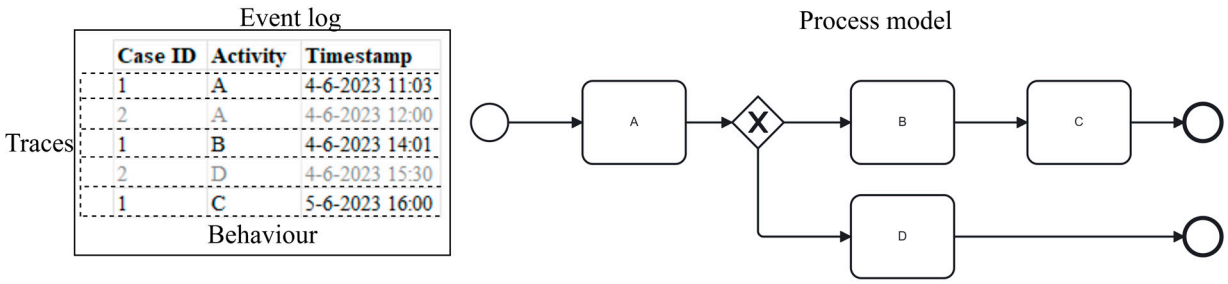


Figure 1. Process quality dimensions concepts (left event log, right process model).

To better address the decision concern, it is important to define fundamental decision-related concepts for the adaptation of the quality dimensions from the process mining domain to the decision mining domain. The first fundamental concept is a fact, which is defined as a piece of information that represents an objective reality or state of affairs [6,17,40]. For example: "*The individual has accumulated five full years of service at their present employer*". The fact in this case is Person Employment Years. The second fundamental concept is fact value, which is an element of information (i.e., a data point within a specific context) and an expression of a fact [17]. In this case, the fact value corresponds to the numerical value "five," representing the number of years in the context of the individual's years of service at their current employer. The third fundamental concept is fact type, which refers to the identifier or label assigned to a specific fact [17]. Building forth on the previous example: a fact type for this is EmploymentYears.

The input data between process mining and decision mining are different due to the fact that process mining focuses on sequencing patterns and decision mining on dependency patterns [41]. Therefore, process mining utilizes an event log (1a), as shown in Figure 2. Decision mining utilizes a decision log (2a), which is a log file containing detailed information about the decisions that have been executed. The decision log consists of an ID, a timestamp, a combination of fact types and fact values, as shown in Figure 2 [10,12].

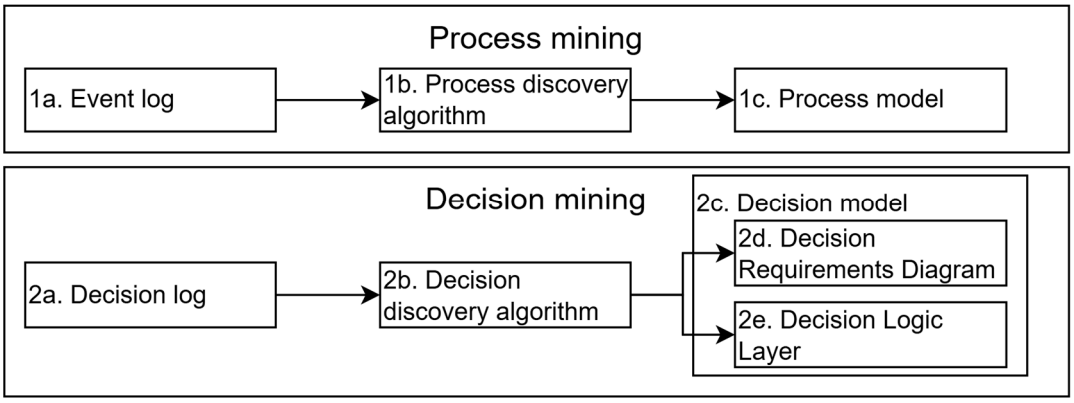


Figure 2. Process and decision discovery comparison.

Due to the different input data (and resulting output data), the discovery algorithm is different in nature. The process discovery algorithm (1b) must handle event data from the event log and should provide a process model (1c) as its resulting model. The same argument could be used for the decision discovery algorithm (2b). Different input- and output data result in a different discovery algorithm compared to a process discovery algorithm. This decision discovery algorithm should handle a decision log as input data

and should provide a decision model as output data. A decision model is a (graphical) representation of a combination of decisions and its underlying decision logic. Examples are the Decision Model and Notation (DMN) [4], The Decision Model (TDM) [17], and Semantics Of Business Vocabulary And Business Rules (SBVRs) [42].

The separation in the decision model between a decision requirements diagram and a decision logic layer is not unique to just the decision modeling language of DMN, for this is also the case in other decision modeling languages, such as TDM [17] and SBVR [42].

The mapping of the decision discovery algorithms on decision quality dimensions in the conformance checking phase involves the utilization of both a decision log (2a) and a decision model (2c) in order to evaluate adherence to a set of constraints specific to each quality dimension, as shown in Figure 2.

3. Research Methodology

This study employs a design science research approach [43] to adapt the precision, fitness, generalization, and simplicity quality dimensions from process mining to the decision mining domain. Following the exaptation strategy proposed by Hevner and Gregor [30], this research builds upon existing process mining methodologies and recontextualizes them for decision mining. The method involves a structured, three-step process: (1) defining the quality dimensions for decision mining, (2) developing metrics and functions for precision, fitness, generalization, and simplicity, and (3) evaluating these metrics through a running example.

3.1. Defining Decision Quality Dimensions

Precision, fitness, generalization, and simplicity are defined based on their roles in assessing the performance of decision discovery algorithms:

- **Precision**: Measures the extent to which the behavior exhibited by the discovered decision model is observed in the decision log. It checks for over-specification by identifying "extra" fact types and fact values not present in the decision log.
- **Fitness**: Evaluates how well the discovered decision model replicates the cases in the decision log, focusing on under-specification by examining "unused" fact types and fact values which are in the decision log.
- **Generalization**: Measures whether the discovered model can apply its learned patterns to future or unseen cases (data not used during training). It prevents overfitting by validating the model on a separate test set.
- **Simplicity**: Assesses the structural complexity of the discovered decision model. A simple model is easier to understand, validate, and maintain by human actors.

3.2. Development of Metrics and Functions

The technical implementation of the precision, fitness, generalization, and simplicity functions is based on comparing sets of fact types and fact values between the decision model and the decision log:

- **Fitness Function**: Calculates the proportion of used fact- types and values relative to the total observed in the decision log. A high fitness score indicates a good match between the model and the actual decision-making data.
- **Precision Function**: Determines the ratio of relevant fact types and values in the decision model to those present in the decision log, indicating how accurately the model reproduces the observed decision behavior.
- **Generalization Function**: Evaluates the model's ability to reproduce unseen cases by measuring its fitness on an out-of-sample dataset.

- **Simplicity Function**: Calculates the structural simplicity of the decision model by comparing its observed complexity to a predefined domain threshold. The resulting percentage expresses the inverse of normalized complexity, where higher values indicate a simpler and more comprehensible model.

3.3. Evaluating Metrics Through Running Example

A running example using DMN is utilized to demonstrate the practical application of the proposed metrics, as follows:

- The example involves a decision model for granting environmental permits based on compliance with emission standards.
- The discovered decision model is compared against the decision log to calculate precision, fitness, generalization, and simplicity scores.

4. Running Example

In the following section, we introduce a running example to illustrate the calculation of the proposed decision mining metrics: fitness, precision, generalization, and simplicity. This example is modeled using the DMN standard [4], which provides a graphical language for specifying decision models and their underlying logic. DMN consists of two layers, the decision requirements diagram (DRD) and the decision logic layer. The DRD contains the dependencies between decisions and the decision logic layer contains the decision logic of the decisions in the DRD.

Figure 3 presents the DRD for the decision of granting an environmental permit, structured into two views: the Existing Decision Model (left) and the Discovered Decision Model (right). Both diagrams center on the decision: 'Grant Environmental Permit'. This high-level decision is shown to be dependent on inputs: the outcome of the 'Environmental Impact Assessment' and the result of the 'Emission Compliance Check' (or 'Emission Conformant Check' in the discovered version). The intermediate decision, 'Emission Compliance Check', is, in turn, dependent on two input data elements: 'Type of Industry' and 'Emission Levels'.

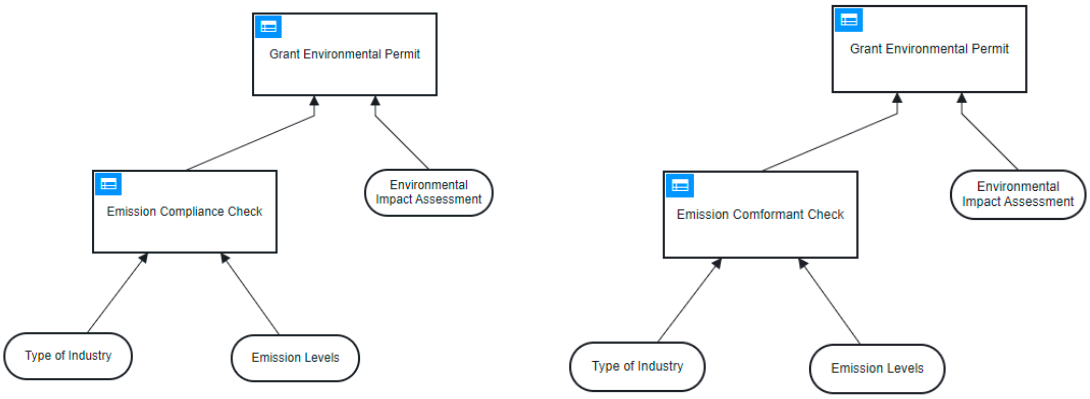


Figure 3. Running example DRD (**left** existing decision model, **right** discovered).

Figure 4 details the Decision Logic for the intermediate sub-decision, the 'Emission Compliance Check'. This figure presents the existing decision log (left, representing the existing decision log) with the discovered decision log (right, the output generated by the decision mining algorithm). An additional rule is discovered (rule 3) in the discovered decision log as compared to the existing decision log.

Emission Compliance Check

Hit Policy: Unique

	When Type of Industry string	And Emission Levels integer	Then Emission High Compliance Check string
1	"Green"	< 250	"High Compliance"
2	"Clean"	[250..350]	"Acceptable Compliance"
3	"Heavy"	> 500	"Non-Compliant"
4	"Green"	-	"High Compliance"

Emission Conformant Check

Hit Policy: Unique

	When Type of Industry string	And Emission Levels integer	Then Emission Conformant Check string
1	"Green"	< 250	"High Compliance"
2	"Clean"	[250..350]	"Acceptable Compliance"
3	"Moderate"	> 500	"Moderate Compliance"
4	"Heavy"	> 500	"Non-Compliant"
5	"Green"	-	"High Compliance"

Figure 4. Running example decision logic Emission compliance check (left decision log, right discovered decision logic).

The running example is utilized for the calculation of the fitness, precision, generalization, and simplicity metrics as discussed in the following section.

5. Precision, Fitness, Generalization, and Simplicity Demonstration

In this section, we elaborate in more detail on how the quality dimensions of precision, fitness, generalization, and simplicity are implemented in order to be used for decision mining. The flowchart as presented in Figure 5 illustrates a high-level overview of the integration of precision, fitness, generalization, and simplicity metrics into the decision mining process. It provides a step-by-step overview of how decision discovery algorithms and decision models are evaluated and used for the improvement of decision models.

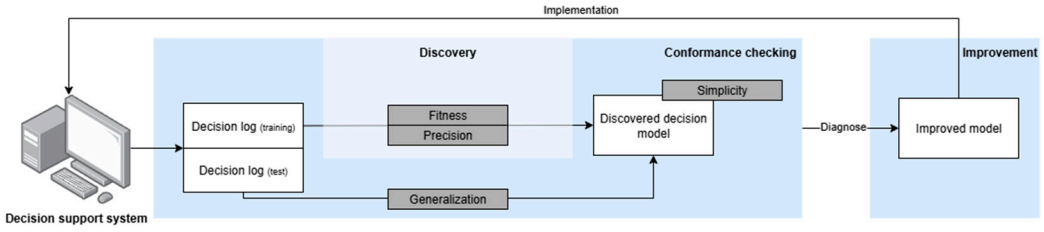


Figure 5. Integration of precision, fitness, generalization, and simplicity. Metrics in the decision mining process.

The process begins with the collection of decision logs, which serve as the primary input for generating decision models through decision discovery algorithms. Once a decision model is discovered, precision, fitness, generalization, and simplicity quality dimensions are applied to assess its quality.

Precision measures how accurately the decision model aligns with the observed behavior in the decision log (as depicted in Figure 5), highlighting any unnecessary complexity or over-specification.

Fitness evaluates how well the decision model replicates all valid cases from the decision log (as depicted in Figure 5), identifying gaps or under-specification in the model.

Generalization assesses the model's ability to reproduce unseen or future decision cases (as depicted in Figure 5), ensuring that the discovered model performs beyond the training data and is not overfitted to only the training data.

Simplicity captures the level of simplicity of the decision model by quantifying the absence of unnecessary complexity (as depicted in Figure 5), indicating how easily the model can be understood by human actors.

In the context of decision mining, the optimal solution is defined not merely by achieving perfect fitness, precision, generalization, or simplicity in isolation, but by finding the balance between them. The optimal decision model is one that exhibits high fitness without compromising the other quality dimensions.

Based on these quality dimensions, a diagnosis is provided and used as input for an improved decision model. Finally, the decision model could be implemented in a decision support system.

The following sections focus on a more in-depth description of the technical implementation of the fitness, precision, generalization, and simplicity functions. First, the formal definitions are provided, following with the explanation of the fitness, precision, generalization, and simplicity functions. Finally, each function is provided with an evaluation in order to show the actual output for each function.

5.1. Definitions

Let M denote a decision model.

Facttype(x) is a predicate where x is a fact type.

$A = \{a: a \in M \text{ ^ } Facttype(a)\}$. Stands for the set of fact types in the decision model (M).

Let L denote a decision log.

$B = \{b: b \in L \text{ ^ } Facttype(x)\}$. Stands for the set of fact types in the decision log (L).

Factvalue(y) is a predicate where y is a fact value.

$C = \{e: e \in M \text{ ^ } Factvalue(e)\}$. Stands for the set of fact values in the decision model (M).

$D = \{f: f \in L \text{ ^ } Factvalue(f)\}$. Stands for the set of fact values in the decision log (L).

Length(G): Represents the length function, which returns the number of items in a set G. It is crucial for calculating percentages that assess the fitness of the decision model.

U = Unused Fact Types: Represents the set of fact types found in the decision log but not in the decision model, indicating a lack of coverage or anticipation by the decision model.

V = Used Fact Types: Denotes the set of fact types that are found in both the decision log and the decision model, showing alignment and expected use.

W = Unused Fact Values: This set contains fact values that appear in the decision log but are not recognized by the decision model, suggesting potential gaps in the decision model.

X = Used Fact Values: The set of fact values present in both the decision log and the decision model, affirming that the decision model accurately reflects the decision log.

Y = Extra Fact Types: The set of fact types present in the decision model and not in the decision log.

Z = Extra Fact Values: The set of fact values present in the decision model and not in the decision log.

C_M = The observed structural complexity of decision model M, defined in the work of Hasić and Vanthienen [44]. Other methods could be used for determining complexity and, therefore, we do not determine a specific method and only refer to the work of Hasić and Vanthienen [44] as an example.

C_{max} = The maximum acceptable complexity threshold, set by human actors, used for normalizing the simplicity score.

NCS(M) = expresses the relative complexity of a decision model by comparing its observed complexity C_M to a reference threshold C_{max}. It ranges from 0 (minimal complexity) to 1 (above-threshold complexity).

L_{train} = A partition of the decision log L into a training set *(L_{train})*.

L_{test} = A partition of the decision log a test set *(L_{test})*.

$Fitness_{train}(M)$ = The fitness of decision model M on the training log L_{train}.

$Fitness_{test}(M)$ = The fitness of decision model M on the test log L_{test}.

5.2. Fitness

The precision, fitness, generalization, and simplicity implementations both compare the decision model and the decision log. Fitness assesses the degree to which the discovered decision model can accurately replicate the cases in the decision log. The concepts of overfitting and underfitting arise when dealing with the case of assessing a decision model on fitness. In an underfitting model, we might include too few input variables or overly simplistic decision logic. The consequences of an underfitting decision model can be substantial. An underfitted model exhibits low generalizability because it fails to capture essential conditions, causing it to produce incorrect outputs for significant portions of the input space when deployed. Moreover, such a model cannot adequately explain the causal mechanisms present in the original decision log, resulting in a representation that does not reliably reflect the organization's actual decision-making behavior. In an overfitting model, we might include too much decision logic, leading to a model that performs very well on the training data but poorly on new, unseen data. The fitness implementation checks for two constraints are as follows: to assess whether there are fact types and fact values in the decision log that are missing in the decision model (underfitting) and whether there are extra fact types and fact values used in the decision model, which are not there in the decision log. The fitness function evaluates the congruence between a decision model and a decision log, if a decision model and decision log are in perfect balance, and if all fact types and fact values in the decision log are represented in the decision model.

The function generates a dictionary containing the fitness percentage and information regarding unused and new facts (fact types and fact values). We propose the following procedure to calculate the fitness of the decision model as follows:

1. **Classify Fact Types as Used or Unused**: Iterates over each fact type used in the decision model or not. Fact types not in the decision model are added to the unused set U, and those in both the log and model are added to the used set V.

Unused fact types:

$$U = B - A$$

Used fact types:

$$V = A \cap B$$

2. **Classify Fact Values as Used or Unused**: Similar to fact types, this step classifies each fact value from the log as either unused (W) or used (X) based on its presence in the model.

Unused fact values:

$$W = D - C$$

Used fact values:

$$X = C \cap D$$

3. **Calculate Fitness Percentages**: Calculates the percentage of used fact types and fact values relative to the total fact types and fact values observed in the decision log, providing a measure of how well the decision model fits the real-world data represented in the decision log.

$$\text{FactTypesPercentage} = \frac{\text{Length}(V)}{\text{Length}(V) + \text{Length}(U)} \times 100$$

$$\text{FactValuesPercentage} = \frac{\text{Length}(X)}{\text{Length}(X) + \text{Length}(W)} \times 100$$

$$\mathit{Fitness} = \frac{\text{FactTypesPercentage} + \text{FactValuesPercentage}}{2}$$

Fitness testcase

As described in the Fitness function, the output of the function is as follows: a fitness percentage, a list of unused fact types, a list of used fact types, a list of unused fact values, and used fact values. The following fitness-related information is a result of the comparison of the running example through the fitness function:

Fitness Percentage: **86.7%**

Unused Fact Types: Emission Conformant Check

Used Fact Types: Type of Industry, Emission Levels, Environmental Impact Assessment, and Grant Environmental Permit

Unused Fact Values: Moderate

Used Fact Values: Green, <250, High Compliance, Positive, TRUE, Clean, [250..350], Acceptable Compliance, >500, Moderate Compliance, Neutral, FALSE, Heavy, and Non-Compliant

The fitness outcome indicates that the discovered model has an 86.7% fitness percentage due to the 'emission conformant check' fact type and 'moderate' fact value being unused.

5.3. Precision

In contrast to fitness, precision quantifies the portion of the discovered decision model's behavior that remains unobserved in the decision log. The precision implementation checks for one constraint. Whether the decision model shows extra fact types and fact values is not directly visible in the decision log. In addition, both quality dimensions check how often a constraint occurs. The precision function estimates the precision of a decision model by comparing it to the decision log. The function generates a dictionary containing the extra fact types and fact values observed in the decision model. We propose the following procedure to calculate the precision of a decision model as follows:

1. **Identify Extra Fact Types**: Evaluates each fact type in the decision model to see if it is not included in the decision log. Fact types not found are considered "extra" as they represent over-specification in the decision model. The logic is as follows:

$$Y = A - B$$

2. **Identify Extra Fact Values**: Similar to identifying extra fact types, it checks the each extra fact values in the decision model against those in the decision log. The fact values not present in the decision log are tagged as "extra", indicating potential over-specification in the decision model. The logic is as follows:

$$Z = C - D$$

3. **Calculate Precision Percentage**: Calculates the percentage of fact types and fact values in the decision model that are actually used according to the decision log. Higher

percentages indicate a decision model more (or to) finely tuned to the decision log's content. The logic is as follows:

$$\text{FactTypesPercentage} = \frac{\text{Length}(A) - \text{Length}(Y)}{\text{Length}(A)} \times 100$$

$$\text{FactValuesPercentage} = \frac{\text{Length}(C) - \text{Length}(Z)}{\text{Length}(C)} \times 100$$

$$\textit{Precision} = \frac{\text{FactTypesPercentage} + \text{FactValuesPercentage}}{2}$$

Precision testcase

As described in the Precision function, the output of the function is as follows: a precision percentage, a list of unused fact types, a list of used fact types, a list of unused fact values, and used fact values. The following precision-related information is a result of the comparison of the running example through the precision function:

Precision percentage: **79.5%**

Extra Fact Types: Emission Compliance Check and Emission High Compliance Check

Extra Fact Values: Low Compliance and Negative

The precision outcome indicates that the discovered model has a 79.5% precision due to that the 'Emission Compliance Check' and 'Emission High Compliance Check' fact types are unused, and 'Low compliance' and 'Negative' fact values are used.

5.4. Generalization

Generalization measures the capacity of a discovered model to represent cases that were not used during the model's training. This concept ensures the model has learned the essential, underlying structure and logic from the training data, rather than merely memorizing its specifics. In practice, this means new, unseen data (or out-of-sample data) should fit within the bounds and structure defined by the model without breaking its logic or requiring unacceptable adjustments. Several steps are necessary to ensure the proper calculation of the generalization metric:

1. **Log Partitioning**: The historical decision log L is partitioned into two disjoint subsets: a training set, L_{train}, which is used to discover the decision model, and a test set, L_{test}, which serves as out-of-sample data to evaluate the model. The formal definition is as follows:

$$L_{train} \cup L_{test} = L$$

$$L_{train} \cap L_{test} = \emptyset$$

2. **Metric Calculation**: The existing fitness metric would then be calculated using the test set. A high fitness score on the test set indicates high generalization. Applying the fitness calculation to the test set (ensuring an out-of-sample dataset) with the discovered model provides a basis for assessing the generalizability of the model.

$$Fitness_{train} = \text{fitness on the training set and decision model } M,$$

$$Fitness_{test} = \text{fitness on the test set and decision model } M.$$

3. **Generalization Calculation**: whether the discovered model can correctly reproduce future or unseen cases. It prevents overfitting by validating the model on a separate test set. Generalization is reported as the fitness on the test set (as seen in Figure 5):

$$\text{Generalization}(M) = \text{Fitness}(M, L_{test})$$

Generalization Test case

Using the running example (environmental permit decision), the log was split 80/20 (training/test). Applying the previously defined Fitness function yields the following:

- $Fitness_{train}$ = 86.7%
- $Fitness_{test}$ = 84.2%

Generalization: 84.2%. The test fitness is slightly lower than the training fitness. The model generalizes reasonably well to unseen cases.

5.5. Simplicity

Simplicity assesses the structural complexity and understandability of the discovered decision model. A simpler model is easier for human actors to validate, maintain, and trust. This quality dimension differs from the other dimensions of fitness, precision, and generalization. While these three dimensions can typically be expressed as percentages and through the presentation of (un)used DMN elements, simplicity is, to some extent, subjective. For this study, we use the work of Hasić and Vanthienen [44] as an example to calculate complexity. These types of metrics assess structural aspects of the DRD and the decision tables, such as the number of decisions, the number of input variables, the depth of dependencies, and the number of rules. Existing approaches lack a normalized measure for which it is necessary to calculate the level of simplicity over 26 different complexity measures. Therefore, this function proposes the following missing measures to calculate simplicity:

1. **Calculating observed structural complexity**: For demonstration purposes, we define and apply a selected set of complexity metrics. These metrics serve to illustrate the proposed approach and can be replaced or extended by other measures as needed. The metrics presented here are conceptually derived from the complexity metrics introduced by Hasić and Vanthienen [45], who provide a broader set of structural and logical indicators for DMN decision models, as follows:

   I(M) = The number of input data elements in the DRD of decision model *M*.
   D(M) = The number of decision nodes in the DRD of decision model *M*.
   DEP(M) = The maximum dependency depth in *M*, i.e., the length of the longest path from an input node to the top decision.
   R(M) = The total number of rules across all decision tables in *M*.
   C(M) = The total number of conditions across all decision table in *M*.
   The total structural complexity could be calculated as follows:

$$C_M = I(M) + D(M) + DEP(M) + R(M) + C(M)$$

2. **Setting domain-expert threshold:** In the simplicity function, a threshold should be specified (when is a model simple?) by a domain-expert. This threshold is necessary to indicate simplicity and not just complexity.

Therefore, the following definition of the domain-expert threshold is introduced, as follows:

C_{max} = The maximum acceptable complexity threshold, set by human actors, used for normalizing the simplicity score.

3. **Calculating Normalized Complexity Score:** Several complexity metrics for decision models have been proposed in the literature, ranging from structural measures (e.g., number of decisions, dependency depth) to logical measures (e.g., number of rules, number of input conditions). These metrics, however, operate on different scales, which makes direct comparison and aggregation problematic. To address this imbal-

ance, we introduce a Normalized Complexity Score (NCS). The NCS expresses the relative complexity of a decision model by comparing its observed complexity C_M to a reference threshold C_{max}. It ranges from 0 (minimal complexity) to 1 (maximum or above-threshold complexity). The Normalized Complexity Score (NCS) of M is defined as follows:

$$\text{NCS}(M) = \min\left(1 \frac{C_M}{C_{max}}\right)$$

4. **Calculate Simplicity percentage**: the NCS of M is translated to a Simplicity percentage through the following calculation:

$$\text{Simplicity}(M) = 100 \times (1 - \text{NCS}(M))$$

A low percentage indicates that the model is complex, while a high percentage indicates that the model is simple. When the observed complexity C_M exceeds the expert-defined threshold C_{max}, the simplicity score is 0% (as the model is considered maximally complex). Conversely, the greater the gap between C_M and C_{max} (with $C_M < C_{max}$), the higher the resulting simplicity percentage.

The challenge in applying simplicity as a quality dimension lies in defining appropriate thresholds. In other words, at what point, for example, does a rule count of 25 represent a simple or a complex model. Especially in the decision domain, this should be conducted by a domain- or subject-matter expert.

Simplicity testcase

As described in the simplicity function, the output is a simplicity percentage (high percentage = a simple model, low percentage = Complex model). In the simplicity function, a threshold should be specified (when is a model simple?) by a human actor (for the testcase this is threshold is 25). Using the running example (Figures 3 and 4), the following complexity metrics could be calculated:

Complexity metrics:
I(M) = 3
D(M) = 2
DEP(M) = 2
R(M) = 5
C(M) = 2
Human actor set threshold (C_{max}): 25
Simplicity percentage: **44%**

6. Discussion

The goal of this study is to identify the differences between precision, fitness, generalization, and simplicity in the context of process mining and decision mining and how to bridge this gap so that precision, fitness, generalization, and simplicity can be defined in relation to decision models.

Previous studies, such as Mannhardt et al. [36] and Scheibel and Rinderle-Ma [37], have addressed decision mining methodologies but often lacked a concrete evaluation framework for discovery algorithms. This study contributes to the field by providing a structured approach to evaluate decision models, filling a gap in the literature identified by Banham et al. [31] and Wais and Rinderle-Ma [32], who called for metrics beyond simple accuracy. Unlike prior research, this study specifically adapts precision, fitness, generalization, and simplicity metrics from process mining to decision mining, accounting for unique characteristics such as fact types and fact values. Baseline algorithms in decision mining, such as ID3- or C4.5-based decision tree learning [45], are typically optimized for prediction rather than model quality in terms of representativeness or conformance. Our

precision, fitness, generalization, and simplicity metrics instead offer model-level insights, supporting decision support system developers in selecting or refining decision logic that balances complexity and behavioral accuracy.

The fitness, precision, generalization, and simplicity functions are proposed, which demonstrate the unique aspects that should be taken into consideration when utilizing the quality dimensions of process mining in the decision mining domain. The current study has certain limitations that need to be addressed. Firstly, the implementation of precision, fitness, generalization, and simplicity is based on the implementation of these quality dimensions in the field of process mining. However, it is important to note that the concepts and characteristics of decisions differ from those of processes, and therefore, the assumptions that are suitable for process mining may not be applicable to decision mining. For example, decision models consist of two different layers, one focused on the visual representation of dependencies and one focused on decision logic. Process models solely have a process diagram layer. Despite this, it should be emphasized that the implementation of precision, fitness, generalization, and simplicity in this paper considers the unique elements of a decision, such as the decision model (including a decision requirement diagram and decision logic layer) and the presence of fact types and fact values.

Additionally, it is important to mention that the body of knowledge on decision discovery algorithms, especially from a decision viewpoint, is rather scarce [41]. Therefore, validating the decision quality dimensions of decision discovery algorithms is limited by the amount of currently available decision discovery algorithms with a focus on the decision viewpoint. The ability for fully validating the decision quality dimensions is hindered. When decision discovery algorithms are more widely available, future research should focus on validating the quality dimensions through test scenarios and comparing the results more thoroughly.

A key limitation of the study is its reliance on a single running example for demonstration and application purposes. While this example demonstrates the implementation of the precision, fitness, generalization, and simplicity metrics, it inherently limits the generalizability of this study and the proposed metrics. For future research, we refer to the utilization of synthetic test scenarios compared to real-life test scenarios, because it provides the researchers with the ability to control the variables in synthetic scenarios. However, real-life test scenarios should be employed as well to improve generalization towards practice.

Lastly, in the conformance phase, process mining focuses purely on checking the event log and process model in conformance with the process mining quality dimensions. This leaves out diagnosing the discovered process model or the already existing process model on any syntax or semantical violations. Utilizing a process model (discovered or already existing) containing syntax or semantical violations could affect the quality of business operations. The potential ramifications of suboptimal business operations are significant. Moreover, when process or decision models fail to meet the stipulated quality standards, it merely marks the initial phase of a broader issue. Inadequate quality in business processes and decision-making can yield adverse consequences extending beyond the mere failure to satisfy quality requirements. These repercussions may manifest as legal disputes, financial penalties, and the erosion of the organization's overall reputation [46]. Encouragingly, industry professionals are already demonstrating a heightened awareness of this impending hazard, as indicated by recent research findings [12].

For future decision mining research, the possibility to check syntax and semantic violations in existing or discovered decision models should be further studied. This is especially important due to the direct effect of a decision model on decision-making. The contributions of [39,40] are a first step in automatically checking any syntax violations in the

DRD and decision logic layer for the DMN modeling language. Beyond ensuring syntactic and semantic correctness, future research should also consider how quality dimensions can be operationalized for non-technical users. Embedding these metrics into decision modeling environments (e.g., DMN modeling environments) would transform abstract numerical measures into actionable visual indicators, thereby improving usability and supporting informed decision-making by non-technical practitioners.

7. Conclusions

This study aimed to answer the following research question: *How can the precision, fitness, generalization, and simplicity quality dimensions from process mining be adapted to effectively evaluate decision discovery algorithms in the decision mining domain?* To do so, the quality dimensions of process mining were taken as a starting point for the description of the quality dimensions of precision, fitness, generalization, and simplicity. The process mining body of knowledge was used for the operationalization of precision, fitness, generalization, and simplicity in the decision mining domain. The fitness function quantifies the congruence between the decision model and decision log and returns a dictionary containing the fitness percentage and information on unused and new facts. The precision function quantifies the precision of the decision model by comparing it to the decision log and returns a dictionary containing information on the extra behavior observed in the decision model. The generalization function evaluates how well the discovered decision model performs on unseen cases by testing it on an out-of-sample dataset of the decision log. The simplicity function assesses the complexity of the decision model relative to a human actor-defined threshold, returning a percentage that indicates the model's overall simplicity.

From a theoretical viewpoint, this study adds to the body of knowledge on how precision, fitness, generalization, and simplicity can be implemented to be used as decision mining quality dimensions for decision discovery algorithms as part of conformance checking in decision mining. This contribution focuses on developing the conformance checking phase of decision mining and thereby providing the body of knowledge with the first step of evaluating decision discovery algorithms. Additionally, the following differences are identified between the precision, fitness, generalization, and simplicity implementation in the process mining problem space compared to the precision, fitness, generalization, and simplicity in the decision mining problem space:

The difference between a process model and a decision model is taken into consideration when developing decision quality dimensions. A process model consists of one layer and a decision model consists of multiple layers (decision requirement diagram and decision logic layers).

The difference between an event log and decision log is taken into consideration when developing decision quality dimensions. An event log consists of sequencing patterns and a decision log consists of dependency patterns [12].

From a practical viewpoint, this study enables researchers and practitioners to implement decision discovery algorithms with the ability to characterize them on the decision quality dimensions, thereby providing some context to the uncertainty of statistical analyses. Furthermore, this study will help practitioners to explicitly consider relevant public values when designing and implementing a decision mining solution. Additionally, the following difference is identified between the precision, fitness, generalization, and simplicity implementation in the process mining problem space compared to the precision, fitness, generalization, and simplicity in the decision mining problem space:

The importance of the implementation of precision, fitness, generalization, and simplicity in the decision mining problem space could be higher due to its impact. A decision

discovery algorithm, which is evaluated with low precision, has a higher effect on decision-making, and in turn, an organization than a process model with low precision.

Decision mining has the potential to support public values such as transparency, accountability, and fairness [20]. By providing a method to assess the quality of decision models, this study indirectly contributes to these values. A decision model with high precision, fitness, generalization, and simplicity not only aligns with regulatory requirements but also helps organizations build trust with stakeholders by ensuring that decision-making processes are both transparent and consistent. Lastly, for these quality dimensions to be utilized by non-technical users, future work must focus on their integration into a decision support system or an environment that utilizes graphical decision models (such as DMN). This integration is crucial for usability, as it enables the translation of abstract numerical scores into meaningful visual cues (e.g., color-coding, simplified dashboards) that non-technical practitioners can instantly recognize and act upon.

Author Contributions: Conceptualization, S.L. and A.v.d.H.; methodology, S.L., A.v.d.H. and K.S.; validation, S.L. and A.v.d.H.; formal analysis, S.L. and A.v.d.H.; investigation, S.L. and A.v.d.H.; resources, S.L. and A.v.d.H.; data curation, S.L. and A.v.d.H.; writing—original draft preparation, S.L. and A.v.d.H.; writing—review and editing, S.L., A.v.d.H. and K.S.; visualization, S.L. and A.v.d.H.; supervision, K.S. and S.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: No new data was created.

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

Abbreviation

The following abbreviation is used in this manuscript:

DMN	Decision model and notation

References

1. Mircea, M.; Ghilic-Micu, B.; Stoic, M. An agile architecture framework that leverages the strengths of business intelligence, decision management and service orientation. In *Business Intelligence*; Solution for Business Development: London, UK, 2012.
2. Rula, A.; Maurino, A.; Batini, C. *Data and Information Quality: Dimensions, Principles and Techniques*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2016.
3. Chalvatzis, K.J.; Malekpoor, H.; Mishra, N.; Lettice, F.; Choudhary, S. Sustainable resource allocation for power generation: The role of big data in enabling interindustry architectural innovation. *Technol. Forecast. Soc. Change* **2019**, *144*, 381–393. [CrossRef]
4. The Object Management Group (OMG) Decision model and notation (DMN). *Decis. Model Not. DMN* **2019**, *1*. Available online: http://www.omg.org/spec/DMN/ (accessed on 8 October 2025).
5. van der Aalst, W. Three good reasons for using a petri-net-based workflow management system. In *Information and Process Integration in Enterprises*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 161–182.
6. Boyer, J.; Mili, H. Agile business rule development. In *Agile Business Rule Development*; Springer: Berlin/Heidelberg, Germany, 2011.
7. Smit, K. *Organization and Governance of Business Rules Management Capabilities*; Open Universiteit: Heerlen, The Netherlands, 2018.
8. Batoulis, K.; Meyer, A.; Bazhenova, E.; Decker, G.; Weske, M. Extracting decision logic from process models. In Proceedings of the International Conference on Advanced Information Systems Engineering, Stockholm, Sweden, 8–12 June 2015; pp. 349–366.
9. Bazhenova, E.; Weske, M. Deriving decision models from process models by enhanced decision mining. *Lect. Notes Bus. Inf. Process.* **2016**, *256*, 444–457. [CrossRef]
10. Smedt, J.; Broucke, S.K.L.M.; Obregon, J.; Kim, A.; Jung, J.-Y.; Vanthienen, J. Decision mining in a broader context: An overview of the current landscape and future directions. In *Lecture Notes in Business Information Processing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; Volume 281, pp. 197–207.

11. Rozinat, A.; van der Aalst, W.M.P. Decision mining in ProM. In *Business Process Management: 4th International Conference, BPM 2006*; Dustdar, S., Fiadeiro, J.L., Sheth, A.P., Eds.; Springer: Vienna, Austria, 2006; pp. 420–425.
12. Leewis, S. Improving Operational Decision-Making Through Decision Mining. Doctoral Dissertation, Open Universiteit, Heerlen, The Netherlands, 2025.
13. The Object Management Group. Business Process Model and Notation (BPMN) 2.0. 2014. Available online: https://www.omg.org/spec/BPMN (accessed on 8 October 2025).
14. Etikala, V.; Vanthienen, J. An Overview of Methods for Acquiring and Generating Decision Models. In *Knowledge Science, Engineering and Management. KSEM 2021*; Qiu, H., Zhang, C., Fei, Z., Qiu, M., Kung, S.Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12817. [CrossRef]
15. Dasseville, I.; Janssens, L.; Janssens, G.; Vanthienen, J.; Denecker, M. Combining DMN and the knowledge base paradigm for flexible decision enactment. In Proceedings of the RuleML 2016 Challenge, Doctoral Consortium and Industry Track Hosted by the 10th International Web Rule Symposium (RuleML 2016), Stony Brook, NY, USA, 1 January 2016. Available online: https://ceur-ws.org/Vol-1620/paper3.pdf (accessed on 8 October 2025).
16. Devine, P.W.; Srinivasan, C.A.; Zaman, M.S. Importance of data in decision-making. In *Business Intelligence Techniques*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 21–39.
17. Halle, B.; Goldberg, L. *The Decision Model: A Business Logic Framework Linking Business and Technology*; Taylor and Francis Group, LLC: Oxfordshire, UK, 2009.
18. Hirvonen, H. Just accountability structures—A way to promote the safe use of automated decision-making in the public sector. *AI Soc.* **2024**, *39*, 155–167. [CrossRef]
19. Koene, A.; Clifton, C.; Hatada, Y.; Webb, H.; Richardson, R. A governance framework for algorithmic accountability and transparency. *EPRS Eur. Parliam. Res. Serv.* **2019**, 1–124. [CrossRef]
20. Lepri, B.; Oliver, N.; Letouzé, E.; Pentland, A.; Vinck, P. Fair, transparent, and accountable algorithmic decision-making processes. *Philos. Technol.* **2018**, *31*, 611–627. [CrossRef]
21. Levy, K.; Chasalow, K.E.; Riley, S. Algorithms and decision-making in the public sector. *Annu. Rev. Law Soc. Sci.* **2021**, *17*, 309–334. [CrossRef]
22. Veale, M.; Kleek, M.; Binns, R. Fairness and accountability design needs for algorithmic support in high-stakes public sector decision-making. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; pp. 1–14.
23. Wilkinson, S.R.; Aloqalaa, M.; Belhajjame, K.; Crusoe, M.R.; De Paula Kinoshita, B.; Gadelha, L.; Garijo, D.; Gustafsson, O.J.R.; Juty, N.; Kanwal, S.; et al. Applying the FAIR Principles to computational workflows. *Sci. Data* **2025**, *12*, 328. [CrossRef] [PubMed]
24. van der Aalst, W.M.P. *Process Mining*; Springer: Berlin/Heidelberg, Germany, 2011.
25. Buijs, J.C.A.M.; Dongen, B.F.; van der Aalst, W.M.P. On the role of fitness, precision, generalization and simplicity in process discovery. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7565, pp. 305–322. [CrossRef]
26. Buijs, J.C.A.M.; Dongen, B.F.; van der Aalst, W.M.P. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. *Int. J. Coop. Inf. Syst.* **2014**, *23*, 1–39. [CrossRef]
27. van der Aalst, W.M.P.; Adriansyah, A.; Dongen, B.F. Replaying history on process models for conformance checking and performance analysis. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2012**, *2*, 182–192. [CrossRef]
28. Yang, Y.; Zhang, Q.; Zhao, F.; Cheng, Y.; Xie, Q.; Wang, G. Optimal scale combination selection based on genetic algorithm in generalized multi-scale decision systems for classification. *Inf. Sci.* **2025**, *693*, 121685. [CrossRef]
29. Zhang, X.; Huang, Y. Optimal scale selection and knowledge discovery in generalized multi-scale decision tables. *Int. J. Approx. Reason.* **2023**, *161*, 108983. [CrossRef]
30. Hevner, A.R.; Gregor, S. Positioning and presenting design science research for maximum impact. *MIS Q.* **2013**, *37*, 337–355. [CrossRef]
31. Banham, A.; Hofstede, A.H.M.T.; Leemans, S.J.J.; Mannhardt, F.; Andrews, R.; Wynn, M.T. Comparing conformance checking for decision mining: An axiomatic approach. *IEEE Access* **2024**, *12*, 60276–60298. [CrossRef]
32. Wais, B.; Rinderle-Ma, S. Towards a comprehensive evaluation of decision rules and decision mining algorithms beyond accuracy. In Proceedings of the International Conference on Advanced Information Systems Engineering, Limassol, Cyprus, 3–7 June 2024; pp. 403–419.
33. Amini, M.; Bagheri, A.; Delen, D. Discovering injury severity risk factors in automobile crashes: A hybrid explainable AI framework for decision support. *Reliab. Eng. Syst. Saf.* **2022**, *226*, 108720. [CrossRef]
34. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA, USA, 2012.

35. Smedt, J.; Hasić, F.; Broucke, S.K.L.M.; Vanthienen, J. Towards a holistic discovery of decisions in process-aware information systems. In Proceedings of the International Conference on Business Process Management, Barcelona, Spain, 10–15 September 2017; pp. 183–199.
36. Mannhardt, F.; Leoni, M.; Reijers, H.A.; van der Aalst, W.M.P. Decision mining revisited-discovering overlapping rules. In Proceedings of the International Conference on Advanced Information Systems Engineering, Ljubljana, Slovenia, 13–17 June 2016; pp. 377–392.
37. Scheibel, B.; Rinderle-Ma, S. *Decision Mining with Time Series Data Based on Automatic Feature Generation*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 3–18. [CrossRef]
38. van der Aalst, W.M.P. Mediating between modeled and observed behavior: The quest for the "right" process: Keynote. In Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science (RCIS), Paris, France, 29–31 May 2013; Volume 1321, pp. 1–12. [CrossRef]
39. Petri, C.A. Kommunikation mit Automaten (Communication with Automata). Ph.D. Thesis, University of Bonn, Bonn, Germany, 1962.
40. Morgan, T. *Business Rules and Information Systems: Aligning IT with Business Goals*; Addison-Wesley: Boston, MA, USA, 2002.
41. Leewis, S.; Smit, K.; Zoet, M. Putting decision mining into context: A literature study. In *Digital Business Transformation*; Agrifoglio, R., Lamboglia, R., Mancini, D., Ricciardi, F., Eds.; Lecture Notes in Information Systems and Organisation; Springer International Publishing: Cham, Switzerland, 2020; Volume 38, pp. 31–46, ISBN 978-3-030-47354-9.
42. The Object Management Group (OMG). Semantics of Business Vocabulary and Business Rules (SBVR) 2013, 1 4. Available online: https://www.omg.org/spec/SBVR/1.4 (accessed on 8 October 2025).
43. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design science in information systems research. *MIS Q.* **2004**, *1*, 75–105. [CrossRef]
44. Hasić, F.; Vanthienen, J. Complexity metrics for DMN decision models. *Comput. Stand. Interfaces* **2019**, *65*, 15–37. [CrossRef]
45. De Leoni, M.; van der Aalst, W.M.P. Data-aware process mining: Discovering decisions in processes using alignments. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; ACM: Coimbra, Portugal, 2013; pp. 1454–1461.
46. Smit, K.; Zoet, M. Management control system for business rules management. *Int. J. Adv. Syst. Meas. IARIA* **2016**, *9*, 210–219.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.