A Framework for a Data Quality Module in Decision Support Systems: An Application with Smart Grid Time Series

Giulia Rinaldi[1], Fernando Crema Garcia[1], Oscar Mauricio Agudelo[1], Thijs Becker[2,3], Koen Vanthournout[2,3], Willem Mestdagh[1] and Bart De Moor[1]

[1] *ESAT Stadius Center for Dynamical Systems, Signal Processing, and Data Analytics, KU Leuven, 3001 Heverlee, Belgium*

[2] *AMO, Flemish Institute for Technological Research (VITO), Boeretang 200, 2400 Mol, Belgium*

[3] *AMO, EnergyVille, Thor Park 8310, 3600 Genk, Belgium*

Keywords: Data Quality, Decision Support System, Data Cleaning, Quality Indicator.

Abstract: Data quality (DQ) measures data status based on different dimensions. This broad topic was brought to the fore in the ’80s when it was first discussed and studied. A high-quality dataset correlates with good performance in artificial intelligence (AI) algorithms and decision-making processes. Therefore, checking the quality of the data inside a decision support system (DSS) is an essential pre-processing step and is beneficial for improving further analysis. In this paper, a theoretical framework for a DQ module for a DSS is proposed. The framework evaluates the quality status in three stages: as based on the European guidelines, as based on DQ metrics, and as based on checking a subset of data cleaning (DC) problems. Additionally, the framework supports the user in identifying and fixing the DC problems, which speeds up the process. As output, the user receives a DQ report and the DC pipeline to execute to improve the dataset’s quality. An implementation of the framework is illustrated in a proof-of-concept (POC) for an industrial use case. In the POC, an example of the execution of the various framework phases was shown using a public time series dataset containing quarter-hourly consumption profiles of residential electricity customers in Belgium for the year 2016.

1 INTRODUCTION

For most artificial intelligence (AI) projects, the first step an analyst needs to perform is to evaluate the quality of the received data. Based on this investigation, the analyst improves the data quality (DQ) if needed. Therefore, DQ measures the state of the data based on various factors, among which are accuracy, completeness, consistency, and timeliness. The process of fixing the possible dataset issues is called data cleaning (DC), which is composed of pre-processing routines fundamental to guaranteeing the success of further analysis. For instance, it was demonstrated that DQ influences the error rate of machine learning models (Ehrlinger et al., 2019) and the decision-making process (Chengalur-Smith et al., 1999).

Appen, an AI company, submits yearly surveys among data scientists with questions related to Machine Learning (ML) models and data to investigate the state of AI and ML. In the 2018 report, most participants determined that the quality of the training data was their biggest challenge.[1] In 2022, more than 70% of participants declared that they spent at least 30% of their time on pre-processing tasks [2].

Assisting an analyst during this first phase can help to speed up a critical process that requires time and attention. Common practice reveals the tendency of analysts to do their own custom DC process from scratch, even though another analyst had already cleaned previously. This repetition of work could be seen as a waste of resources. Organizing the job in a way that researchers can evaluate and apply the same data-cleaning process to the datasets would allow them to save time.

This paper proposes a module to handle DQ inside a decision support system (DSS). A DSS is a computer software tool that aids a user in managing and making decisions on the data. The defined module analyzes the input to extract useful information during the data-cleaning process. It checks the quality of the inputs in three processes:

- EU guidelines process: the focus is mainly on the

[1] https://visit.figure-eight.com/rs/416-ZBE-142/images/Data-Scientist-Report.pdf

[2] https://appen.com/blog/2022-state-of-ai-machine-learning-report/

Rinaldi, G., Crema Garcia, F., Agudelo, O., Becker, T., Vanthournout, K., Mestdagh, W. and De Moor, B.
A Framework for a Data Quality Module in Decision Support Systems: An Application with Smart Grid Time Series.
DOI: 10.5220/0011749700003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 1*, pages 443-452
ISBN: 978-989-758-648-4; ISSN: 2184-4992

metadata and reusability of the dataset.

- DQ assessment process: data quality dimensions are checked on the dataset.
- DC assessment process: The system evaluates some common data issues.

Based on the outcomes, the system computes two indicators that summarize the last two processes. The indicators and the analysis details are shown in a final report to submit to the analyst. In the last phase of the DQ module for DSS, the system assists the user in designing the DC process by highlighting some cleaning problems using the information gathered during the previous analysis and using historical experiences stored in a database. The expert decides how to handle them, and each feedback given is saved as historical experience to be available for future analysis on future datasets. Therefore, the framework intends to offer the analyst a way to study the dataset while simultaneously speeding up the pre-processing design pipeline. The framework is then tested on a public dataset, unlike the usual approach, (Sadiq and Indulska, 2017).

Section 2 presents some of the most pertinent research on frameworks and tools for data quality and cleaning. Section 3 explains our framework for a data quality module to be used inside a decision support system. Section 4 illustrates this framework when applied to a use case. Section 5 draws conclusions and delineates possible further research directions.

2 RELATED WORK

DQ started to gain importance during the 1980s when organizations started to rely more on data and data mining. Many dimensions and metrics were defined during the years in various studies and then translated into practical software tools. (Ehrlinger and Wöß, 2022) identified 667 tools developed for DQ tasks. The authors underlined the difficulty of identifying just one definition of DQ. In their work, they give an overview of the concept of data profiling, summarized as the process of collecting metadata, data quality measurements, summarized as the capability of estimating the DQ dimensions, and data cleansing, summarized as the procedure for fixing incorrect data. To evaluate the 667 tools, a requirements catalog consisting of 43 constraints for data profiling, data quality measurements, and continuous data quality monitoring was defined. Only 13 tools respected the constraints and were then described in detail. The authors concluded that there is still a need to research automation in DQ, and the examined tools missed a clear *"declaration and explanation of the performed calculation and algorithms"*.

The same authors proposed an automated data quality monitoring tool (Ehrlinger and Wöß, 2017), which periodically monitors the data quality of heterogeneous data collected by an information system. The employed architecture is comprised of four components. The first is data profiling and quality assessment. During this phase, the metadata and calculated DQ metrics are collected. The user can also provide domain-specific information and additional specification used in the process, such as the monitoring frequency. The results calculated over time are stored in the DQ repository, the second component. The third element is time series analysis which uses well-known algorithms to examine the information in the DQ repository. The last component is visualization, in which the user can plot the collected time series and monitor the obtained results.

The framework proposed by (Oliveira and Oliveira, 2022) monitors DQ using a reliability score. The input of the system is heterogeneous data. Their architecture is based on the scalable publish/subscribe messaging system, Kafka (https://kafka.apache.org/). The main component is the data quality layer which is composed of a plug-in and the data quality analyzer. The latter is the process that produces a data quality index employing rules to analyze the data stored in a JSON (JavaScript Object Notation) file. The plug-in applies and checks the rules on the data. The authors tested the framework to a use case using customer data. In the example, a reliability score was calculated using the DQ dimensions of accuracy, consistency, and completeness. It was used to identify possible outliers.

Like the papers described in this section, our framework intends to monitor the quality status of the data. The main difference is that our framework offers three levels of assessment. The first focuses on the input's format and standard, the second on DQ, and the third assesses common DC problems. In the end, the user has three evaluations to estimate how much time is needed to clean a specific dataset. The tracking of cleaning changes is a new contribution in our framework. It saves the transformations done to improve the data as relevant historical experiences, which will be used for future analysis of similar datasets. In this way, the system learns from the user and speeds up future processes by offering better support. The system identifies the cleaning problems and solves them with the human-in-the-loop. At the end, the user is provided with a report of the raw data and the pre-processing pipeline.

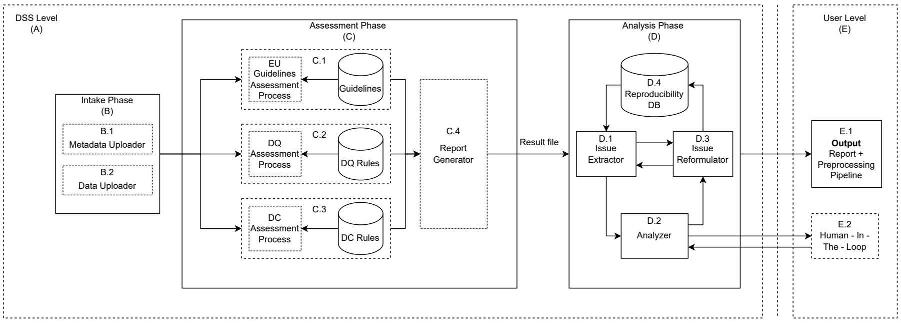


Figure 1: Architecture of the data quality module's framework. It has two levels: DSS Level (A) and User Level (E). DSS Level is composed of three phases: the Intake Phase (B), the Analysis Phase (C), and the Annotation Phase (D). Each of them includes the submodules.

3 DATA QUALITY MODULE'S FRAMEWORK

The framework's architecture is depicted in Figure 1. It has two concept levels: user and DSS. The user level (Fig. 1E) is the supposed client of the DSS, while the DSS level (Fig. 1A) is on the server side.

The DSS level starts with the Intake phase (Fig. 1B), and it describes how the user should upload the metadata and the data into the DSS. There are two subphases: metadata uploader (Fig. 1B.1) and data uploader (Fig. 1B.2). The first aims to define the metadata input related to the data, which should include the description and source of the data. On the other hand, the data uploader determines the architecture constraints such as the data format – for example, CSV (Comma Separated Variable) file for tabular and time series data or tiff (Tag Image File Format) for images. In the proof-of-concept (POC) described in Section 4, this last option is not implemented but is part of future research. The metadata and the data are the input for the next phase, the assessment phase (Fig. 1C), which evaluates the user's input in three stages composed of an assessment process and a database containing rules or useful information.

The first stage is the EU guidelines assessment process (Fig. 1C.1), which uses rules taken from the European Guidelines (Data Europa EU, 2021). The European data quality guidelines are recommendations from the European Union to produce high-quality datasets. Their proposal uses the principles of FAIR (Wilkinson et al., 2016), which stands for four data quality dimensions: Findability, Accessibility, Interoperability, and Reusability. The guidelines provide a framework consisting of these dimensions and additional metrics. The EU guidelines assessment process interacts with a database where the rules extracted from the guidelines are stored. The submitted metadata should follow a template defined during the setup of the DSS. The EU guidelines assessment process follows the dimensions:

- Interoperability: The EU guidelines assessment process verifies any encoding issues in reading metadata or opening the data files.
- Findability: The EU guidelines assessment process checks if the metadata at least contains the description of the data, the source of the data, and if it is part of a project, the description of the project. It measures how easy it is to find information and understand the data.
- Accessibility: The EU guidelines assessment process checks how many files the data are divided into and if they are all accessible. Then, it verifies if there are constraints, for example, security constraints, to maintain inside the DSS, such as, for example, "only the admin can access the data."
- Reusability: The EU guidelines assessment process checks the amount of data and any additional rules related to the specific file type. For example, if the file is a CSV, the DSS should check the presence of headers. If the organization using the DSS requires any supplementary standard to be respected, the system will review them at this point.

The result of each phase is saved and shown to the user in a final report.

The second stage of the assessment phase is the DQ assessment process (Fig. 1C.2). The rules related to this stage refer to the data quality metrics:

- Accuracy: The DQ assessment process measures how correct the data are compared to a reference dataset.
- Timeliness: The DQ assessment process evaluates how up-to-date the input dataset is for a task.

- Completeness: The DQ assessment process measures how much information the data carries out. In other words, how much data is not missing from the dataset.
- Consistency: Semantic rules are defined over the data by the user. The DQ assessment process estimates the number of semantic rules which are violated.

Based on the results, a DQ indicator is calculated. This number establishes the data's quality status at any moment. The intent is to provide a summary of the status of the data that is simple to understand but also possible to change depending on the context of the problem. The next steps are a possible guide to calculate this indicator:

1. Calculate quality indicators DQ_i per each dimension $i = \{comp, acc, time, cons\}$ completeness, accuracy, timeliness, and consistency. Each indicator will be defined by the analyst.
2. Assign importance weights (w_i) per each of the data quality dimensions.
   - If the units of DQ_i have the same range, we recommend updating each w_i as:

   $$w_i = \frac{w_i}{\sum_k w_k} \text{ so } \sum_i w_i = 1 \implies \overline{DQ} = \sum_i w_i DQ_i \quad (1)$$

   - If the analyst wants to do the average of the DQ_i, update each w_i as:

   $$w_i = 1 \text{ so } \sum_i w_i = 4 \implies \overline{DQ} = \frac{\sum_i DQ_i}{4} \quad (2)$$

3. Calculate a weighted average of each data quality indicator that we denote as $\overline{DQ}$.

   $$\overline{DQ} = \frac{\sum_i w_i DQ_i}{\sum_i w_i} \quad (3)$$

4. Then, compare this value to a threshold ε with a basic rule:

   if $\overline{DQ} \leq \varepsilon$ *insufficient quality* **else** *sufficient quality*

   This step is recommended but optional.
5. Add results to the result file.

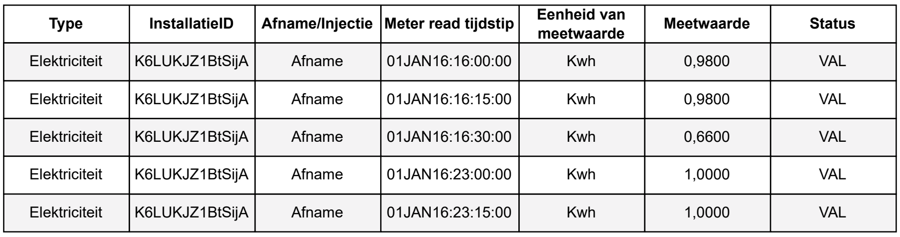
Section 4 presents an example of the DQ indicator.

The final stage of the assessment phase is the DC assessment process (Fig. 1C.3). The rules are based on typical instructions during the data-cleaning procedure. These include

- Time column: The DC assessment process tries to recognize the time column, the start, and the end date, the frequency of the observations, and the presence of gaps or duplicates related to winter/-summer hour time change.
- Single-value columns: The DC assessment process verifies how many columns have only a single value. It should be stored as metadata.
- Types of columns: The DC assessment process tries to assign a specific type to a column based on the values.
- Duplicates: The DC assessment process extracts possible duplicates.
- Missing Values: The DC assessment process verifies if there are any missing values.

As with the previous DQ stage, a DC indicator is calculated based on the result of this stage too. This number aims to provide an idea of the cleaning status of the data. It measures how easily the DSS can clean this dataset without human intervention. This information is linked to the analysis phase because the less information the system defines, the more time is required during the following phase. Under this point of view, the analysts should use it to estimate the time they will probably need to spend during the analysis phase when additional verification will be done. Section 4 illustrates an example of the DC indicator's computation. The framework's modularity offers flexibility. The analyst can decide to skip part of the analysis and can update or change the rules and guidelines to make the DQ module more customizable. After the three assessment stages, the last step in the assessment phase generates a report (Fig. 1C.4) summarizing all the DSS findings.

The last part of the framework is the analysis phase (Fig. 1D) which aims to define the cleaning procedure with the analyst and save it in the reproducibility database. The phase starts with the issue extractor (Fig. 1D.1), a process that formulates potential problems based on the assessment phase report and historical experience already stored in the reproducibility database (Fig. 1D.4). Then, these problems are displayed to the analyst through the analyzer (Fig. 1D.2), which explains the potential issues and shows additional information, such as plots or metadata. At this point, the analyst (Fig. 1E.2) decides whether or not the issue is legitimate and decides on a solution. Finally, the analyst's feedback is passed to the issue reformulator (Fig. 1D.3) and stored in the reproducibility database. This database contains the past problems found by the DSS and how analysts have decided to solve them. It helps the user verify the most common issues and situations that may not be so ordinary. The output of the framework (Fig. 1E.1) is a report on the raw dataset and the pre-processing pipeline designed during the analysis phase.



Type	InstallatieID	Afname/Injectie	Meter read tijdstip	Eenheid van meetwaarde	Meetwaarde	Status
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:00:00	Kwh	0,9800	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:15:00	Kwh	0,9800	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:16:30:00	Kwh	0,6600	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:23:00:00	Kwh	1,0000	VAL
Elektriciteit	K6LUKJZ1BtSijA	Afname	01JAN16:23:15:00	Kwh	1,0000	VAL

Figure 2: Extract of the data contained in the CSV file and used as the proof-of-concept. The column “Type” contains what type of measurements were collected, “InstallatieID” indicates the digital meter id; “Afname/Injectie” indicates if the energy is off-taken or injected; “Meter read tijdstip” represents the timestamps; “Eenheid van meetwaarde” is the unit of measure; “Meetwaarde” collects the measurements; “Status” indicates whether the data is valid or not.

4 PROOF-OF-CONCEPT: INDUSTRIAL APPLICATION

To test the proposed framework, a POC was developed. The programming language used is python, [3] and the Graphical User Interface (GUI) was implemented using Streamlit. [4]

The POC is a data quality module for a data-driven decision support system used in a research organization. The employed dataset is a public industrial time series dataset. The publisher is Fluvius, a Belgian distribution system operator. The data represents the quarter-hourly consumption profiles of residential electricity customers in Belgium for the 2016 calendar year. The use case from which this example was extracted relates to how to use AI to manage low voltage grids more efficiently. The experiment presented in this section is the result of a collaboration with the researchers of the Smart grid use case of the AI Flanders Research Program. [5]

4.1 Intake Phase

After downloading the dataset from the Fluvius website,[6] the user will have a zip file containing a CSV (*READING*_2016.*CSV*) with the measurement data and a XLSX file (1_04-*werkelijkeverbruiksprofielenhuishoudelijke-klantenelektriciteit*2016*Legende.xlsx*) containing the metadata. The presented dataset is in Dutch. The first step is to upload the data and the metadata into the DSS. The intake module receives the CSV, a small extract of it is presented in Figure 2, and then the user needs to deliver the metadata. To do so, the system provides a template to fill in. The user is not obliged to complete all the fields to proceed with the analysis, however, the more information provided, the higher the final score.

4.1.1 Data and Metadata Uploaders

The template for the metadata designed for the POC has two parts: General information and Data information. General information is for the future usage of the submitted data. It consists of the following:

- **The Project’s Name**: Low Voltage Grid - Forecasting Energy Consumption
- **The Domain**: Industrial
- **The Problem Statement**: The visibility on the low voltage (LV) distribution grids throughout Europe is limited: the layout of the grids is only partially known, and measurements are limited. In the past, this was acceptable, as we employed a “fit and forget” strategy: install (overdimensioned) cables with sufficient capacity to cover all demand peaks. Today, this solution is not an option. Now, the alternative is to develop technology to use the installed capacity more optimally by operating our grids closer to their limit and technically supporting measures to mitigate the impact of the energy transition.
- **The Research Goal**: To run long-term forecasts algorithms for all potential evolutions in the grid use (Botman et al., 2022), (Soenen et al., 2023).

Data information is more related to how the data was retrieved. In the example template, the requested information is:

- **The Data Type**: Time Series
- **The Source**: Fluvius Open dataset
- **The Data Description**: The dataset contains 100 timeseries with quarter-hourly offtake and injection measurements of Low-Voltage (LV) grid

[3] https://www.python.org

[4] https://streamlit.io

[5] https://www.flandersairesearch.be/en

[6] https : / / opendata.fluvius.be / explore / dataset / 1_04 - werkelijke - verbruiksprofielen - huishoudelijke - klanten - elektriciteit/information/

connections. These anonymized measurements were obtained in a digital meter proof-of-concept project in 2016, which was carried out in a pilot area of the Belgian territory. This dataset only includes the digital meters for which more than 98% of their readings were validated. Given that the total amount of expected measurements is equal to 96 (readings/day) x 366 (days/year) = 35136 (readings/year), the selected digital meters in this dataset have more than 35136 x 0.98 =34433 validated readings.

Then, there was a drop-off space where the data file had to be added. The last step of this phase is to move the CSV file (Figure 2) with the data to the proper directory and to write the metadata in a JSON file (Listing 1). Then, the system starts the next phase-the assessment phase.

Listing 1: Extract of the JSON file saved during the upload of the metadata in the execution of Proof-of-Concept.

```
{”general”: {
        ”name”: ”Low Voltage Grid”,
        ”problem_statemnt”: ”...”},
    ”data”: {
        ”type”: ”Time Series”,
        ”description”: ”...”}}
```

4.2 Assessment Phase

The assessment phase accesses the CSV and the JSON files, the outputs of the previous phase (Section 4.1). Then, the three processes of this phase begin. The system automatically executes this phase following the configuration setup by the user.

4.2.1 EU Guidelines Assessment Process

The EU guidelines assessment process examines the inputs, comparing them with rules extracted from the European Guidelines (Data Europa EU, 2021). The four dimensions considered in this step are:

- Interoperability: The DSS checks the presence of an encoding issue inside the data and metadata. In the POC, the python library cChardet [7] was used for this task. The system did not find any encoding issue; the results are illustrated in Table 1. The outcome was stored in the JSON result file to be later translated into the final report.
- Findability: The DSS prepares the working space for the Intake Phase's output. The system checks if the project already exists. If so, the metadata file is moved to the already-existing project's working space as a subproject. If not, a new working space is created. In the POC, the working space was a file system directory composed of a general directory called “Working Space”. It contained the project directory, which in turn contained the directories for the data, the metadata, and the future result. From the metadata file, the data's description was examined using text quality indicators implemented in python (Kiefer, 2019). Specifically, the POC implemented the number of Spelling Errors, [8] Lexical Diversity, [9] number of Ungrammatical Sentences, [10] and Average Sentence Length. All the results on the metadata were saved in the JSON result file. The results obtained by the POC are shown in Table 2. They give an indication of the clarity of the metadata input. In this case, for example, the lexical diversity, in which the maximum is one, is quite low. This means that there are many repetitions of the same words. This analysis is important for the reusability of the dataset by other users.
- Accessibility: At this point, the DSS has moved the data file to the working space. During this step, as described in Section 3, the system checks for any possible security constraints, such as limited clearance access for certain personnel. In the POC, there were no introduced constraints.
- Reusability: The last step is to verify the data itself. In the POC, the system read the content of the CSV file (Figure 2) without any problems. The content was loaded as an object called a dataframe.[11] If this operation had caused an error, the process would have read how to mitigate the situation from its associate database. For example, a mitigation action could be asked for the user's involvement. Having the dataframe, the DSS checked the number of samples, the presence of headers, and any other constraints the user might have required. The success or failure of each instruction was saved in the JSON result file.

Table 1: Results obtained from the EU guidelines assessment process: Interoperability Check.

CSV containing Data	JSON containing Metadata
'encoding': 'ASCII'	'encoding': 'ASCII'
'confidence': 1.0	'confidence': 1.0

[7] https://pypi.org/project/cchardet/

[8] https://pypi.org/project/pyenchant/

[9] https://github.com/kieferca/quality-indicators-for-text

[10] https://pypi.org/project/language-tool-python/

[11] https : / / pandas.pydata.org / docs / reference / api / pandas.DataFrame.html

Table 2: Results obtained from the EU guidelines assessment process after the analysis of data's description.

Rule	Result
# Spelling Error	4
Lexical Diversity	~ 0.091
# Ungrammatical Sentences	2
Avg Sentence Length	23.5

An extract of the JSON file result, including the output of the accessibility and reusability, is shown in the Listing 2. To summarize the EU guidelines assessment process, the inputs are the metadata file (JSON file), the data file (CSV file), and the output is the JSON result file, containing the results of each subprocess.

4.2.2 Data Quality Assessment Process

During the second process, the DSS examines the quality of the data contained in the CSV input in depth. The analyst input $w_{comp} = 0.3$, $w_{acc} = 0$, $w_{time} = 0.3$ and $w_{cons} = 0.4$ into the DSS. Finally, the threshold ε was set to 0.8.

The DQ_i indicators were calculated as follow:

- Completeness: In Section 3, this metric (DQ_{comp}) evaluates how much information the dataset has. So, in the POC, it was calculated as

$$\frac{\#\,Expected\ values\ -\ \#\,Missing\ values}{\#\,Expected\ values}. \tag{4}$$

  Numerically, it was $DQ_{comp} = \frac{35136-20078}{35136} = 0,43$.

- Accuracy: In the POC, this metric (DQ_{acc}) was not calculated. Accuracy measures the distance between the input data and a reference dataset. Having a reference dataset is not always possible because, among other reasons, it requires a lot of time for the expert to generate a dataset containing what is desirable. Moreover, the representativeness of a reference dataset loses its meaning fast. Energy consumption data depends on many factors; among them, there are human habits and the employment of new technology, such as heat pumps and electric vehicles. The reference dataset should reflect these continuous changes, causing frequent updates that are not sustainable. Therefore, the system, not finding the reference dataset, skipped this evaluation.

- Timeliness: In Section 3, this metric (DQ_{time}) was defined as a measurement to evaluate how up-to-date the input is. In the POC, the data available was from 2016 because it is hard to have values from a recent period. Recent studies, (Eurostat, 2020), underlined that the electricity consumption by households in Belgium had decreased by 6.7% compared to the usage in 2010. On average, there was a decrease of 1.2% in the European Union. Since the data available was from 2016, the analysts decided to consider the data out-of-date because the percentages are significant, however still relevant for the considered use case. Hence, the analyst assigned zero as the timeliness score.

- Consistency: Having the data saved in files is not the most efficient way to work. So, this metric (DQ_{cons}) measures if the user prefers to save the data in an external database and calculates how many semantic rules are violated by the dataset. Figure 3 shows the interaction between the framework's architecture and an external database. The depicted architecture is a simplification of the one shown in Figure 1. The DQ assessment process (Fig. 3C.2) is possible to link to an external centralized database (Fig. 3EXT) defined by the user and containing the data. In the POC, at this point, the DSS established a connection with the external relational database. It retrieved the appropriate table definition and inspected each attribute, comparing the type with the pertinent column in the dataset. If the column were a general object, the system would try to convert the values to the expected type. In Table 3, two example results of this operation are shown. The table contains the names of the column in the dataset (Fig. 2), which are the same inside the external database as attributes, the types of the attribute found in the external database table's definition, the types of the column in the dataset and the result: they match or they do not. Consistency was measured as follows:

$$DQ_{cons} = \frac{\#\,Checked\ attributes - \#\,Failed\ matches}{\#\,Checked\ attributes}. \tag{5}$$

  In numbers, it is $DQ_{cons} = \frac{7-1}{7} = 0,86$.

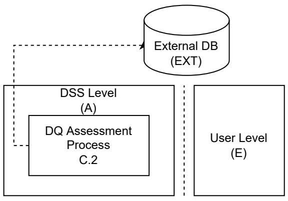


Figure 3: The image shows the interaction between the DQ Assessment Process (C.2) and an External Database (EXT). The depicted architecture is a simplification of Figure 1.

Table 3: Example of results obtained by checking the Consistency of the dataset with the expected types required by the external relational database.

name	attribute:type	column:type	result
InstallatieID	VARCHAR	String	Success
Meetwaarde	FLOAT	String	Fail

The result obtained by this process for each dimension was saved in the JSON result file. The calculations of $\overline{DQ}$ (equation 3), therefore:

$$\overline{DQ} = 0.3 \times 0.43 + 0 \times 0 + 0.3 \times 0 + 0.4 \times 0.86 = 0.45 \qquad (6)$$

As $\overline{DQ} \leq 0.8$ the DSS defines the quality level as insufficient.

4.2.3 Data Cleaning Assessment Process

The DC Assessment Process is the last step before having the final report. During this stage, the DSS inspects some primary data-cleaning issues. The ones implemented in the POC, and already introduced in Section 3, are the following:

- Time Column: The DSS analyzes the column containing the timestamps. In the POC, the first action was to find the range of the considered period, so the start and the end date. Then, reading the hours, the DSS identified the time frequency. Additionally, the system examined the time stamps around March and October with particular attention. At the end of these months, in Europe, there is a change of hours between winter and summer time or vice-versa. This time change of one hour could create duplicated or missing timestamps. The DSS verified the presence of this problem. All these findings were stored in the JSON result file to be double-checked with the user.
- Single-value columns: In the POC, the DSS, iterating on the columns, identified the column with single values. An example is the column “Type” in Figure 2. The value was stored in the JSON Result File, and during the last phase, the system will propose to drop the columns and save the value as metadata.
- Type of columns: This analysis is linked to the CONSISTENCY dimension tested in the previous process. The Consistency test can be performed only if the user provides semantic rules. The DSS automatically tries to recognize the data type if this does not happen during the DC Assessment Process. In the POC, one value of each column was tested to verify if it was a string, boolean, category, int, float, or DateTime.
- Duplicates: In the POC, the DSS calculated the number of duplicated rows found in the system. The user decides how to handle them during the analysis phase, in which all these problems are presented. In the dataframe, the system did not find any duplicated values.
- Missing Values: In the POC, the last examination done by DSS was to verify the presence of missing values. First, the system verified the presence of cells with null values. Then, the system calculated the missing rows using the timestamps column and the information obtained from the metadata. The results are shown in Table 4.

Table 4: Results of the system obtained checking the number of missing values and rows.

missing type	result
null values	0
missing rows	20078

An indicator is calculated at the end of this process, similar to the DQ assessment process (Section 4.2.2). This indicator was computed using Equation 7. In detail, the system counted the total number of all the cleaning problems checked during the DC assessment process, then it calculated the number of actual cleaning problems that the user needs to check in the next phase. The results are shown in Table 5. For example, the test to identify the period range was one, and the system found the answer, so zero problems were needed to be checked in this case. However, during the computation of the missing values, two tests were performed, and one did need human intervention. The final indicator was computed like

$$DC = \frac{\#\,Checked\ problems\ -\ \#Actual\ problems}{\#\,Checked\ problems}. \qquad (7)$$

Then, the obtained number was converted to a percentage. A high score indicator means fewer problems to check during the next phase. In the presented scenario, $DC\ Indicator = \frac{17-7}{17} \times 100 = 37\%$.

Table 5: Collected information to compute the DC indicator. The first column corresponds to the kind of problem, the second represents the number of actual cleaning problems found, and the third is the number of checked problems performed.

	actual problems	checked problems
id period range	0	1
id time frequency	0	1
winter/summer	2	2
single values per each column	3	7
type check	1	3
duplicates	0	1
missing values	1	2

4.2.4 Report Generator

The assessment phase concludes with the generation of the report, summarizing all the findings gathered during the three processes. The input of this block is the JSON result file, an extract of the one produced by the POC is depicted in Listing 2.

This module’s subphase aims to allow the user to review the results and, if needed, download them in a report.

Listing 2: Extract of the JSON result file containing the findings of the three assessment process developed in the Proof-of-concept's. It illustrates the result gathered during the EU Guidelines Process.

```
{”EU”: {
   ”INTEROPERABILITY”:
         {”Data”:
            {”encoding”:”ASCII”,
            ”confidence”:1.0},
         ”Metadata”: {”...”}},
   ”FINDABILITY”:
         {”Sub_project”:”NO”,
          ”Location_workspace”:”...”,
          ”metadata_indicator”:  ,
          ”metadata_details”: {
             ”Spelling”:”2”,
             ”...”}},
   ”ACCESSIBILITY”: {”Visible”:”All”,
         ”Modificable”:”All”},
   ”REUSABILITY”: {”data”:”Well Formed”}}
```

The report should summarize each step executed by the DSS until now and contain all the relevant information related to the project.

In the POC, a template was designed and filled with the obtained results. The DSS started generating the final document filling in the project's name. Then, it added the project's description, the data's description, and an extract of the dataset, similar to Figure 2. After this, the DQ and DC indicators were shown as the first result. Following that, the metadata evaluation was introduced as in Table 2 during the EU guidelines assessment process. All the other results were organized in distinct sections, each for a particular step in the process similar to the pattern used in Section 4.2.

4.3 Analysis Phase

The last phase in the proposed DSS module framework is the analysis phase. This phase is designed in a semi-automatic fashion. The user is involved in every step and actively participates in the decision-making process. The system supports the user in gathering information and underlining potential decisions to make. The input is the JSON result file acquired from the assessment phase; the output is a series of pre-processing steps designed to improve the quality of the data. Figure 1D shows the processes composing this phase.

The issue extractor has the job of formulating the possible cleaning problems based on the JSON result file and based on the experiences collected inside the reproducibility database. In the POC, this database was designed as a relational database. It was used to store all the instructions executed by the user to investigate and solve a cleaning problem. The information stored inside the database is the following: the "id", which represented the identification of the historical record, the "datatype", which was the type of data used for that specific record, the "project", which was the name of the analyzed project, the "problem", which was the name of the data cleaning issue the DSS and the user found, the "solution", which was the code performed by the user to solve the cleaning problem, and the "info", which was the code used by the user to visualize the data and investigate the problem. All the attributes are strings, except "id" which is an integer, and "solution" and "info" which are JSON. The issue extractor took from the JSON result file the data type, Time Series, and the first problem to check with the user, for example, "Missing Values." Then, the process searched for entries in the reproducibility database with datatype=Time Series and problem=Missing Values; if there were any past experiences, all the information was passed to the analyzer.

The analyzer is the interface between the system and the user. In the POC, the GUI was divided into four spaces:

1. Visualization space was where the DSS results of the under-examination problem were shown. The DSS showed the Missing Values test results during the DC assessment process, Section 4.2.3.
2. Experiences space was where the DSS showed the experiences found in the reproducibility database, if any. Firstly, it offered the possibility of visualizing the analysis done. Therefore, the system proposed the solutions.
3. Study space was where the user performed instructions to investigate the problem. The analyst could start from a code proposed in the previous space or propose a personal investigation.
4. Solution space was where the user could accept a past solution or write a new one. For example, the proposed solution could handle the missing data using the interpolation technique. The user could refuse it and instead use the personal solution of filling in the missing values by copying the previous row's values.

When the user was satisfied with the solution, all the new feedback was passed to the issue reformulator. This process had the job of creating the entry for the reproducibility database. It translated the python code written in the study space and the solution space into two JSON objects. Then, it gathered all the information to complete the input query and sent it to the database. To conclude the cycle, the issue reformulator process notified the issue extractor of the operation's success, and so a new problem is passed to the analyzer.

After showing all the problems examined by the DSS during the assessment phase, the system checks the presence of different problems related to the datatype inside the reproducibility database. If any, these are shown to the user. This step helps review problems that, maybe, the analyst has not thought about. In the end, the user could propose new problems to test, and they would be saved as new experiences in the database.

5 CONCLUSIONS

DQ and DC are fundamental for any professional working with data. This paper has proposed a framework that helps users to qualitatively better understand their data and to save time in pre-processing it. The framework aims to give a general overview of the data quality status. It computes indicators related to DQ and DC to support the user in estimating the time they need to spend performing the cleaning process. Furthermore, the framework focuses on speeding up the cleaning process, assisting the user during the identification of any problem then providing possible solutions for any cleaning issues. The last part of the paper described the application of the framework in an industrial POC, the low voltage grid. It was shown that some metrics are not always applicable, but the framework can still be relevant. The dataset employed contained public time series data of energy consumption profiles for the 2016 calendar year in Belgium.

In future work, the framework will be tested with different types of datasets and use cases. Another focus will be on how to use historical experiences more effectively and efficiently to better suggest cleaning issues and solutions during the Analysis Phase. Then, the module will be inserted into the design of a data-driven decision support system.

ACKNOWLEDGEMENTS

The authors greatly thank Ms. Lola Botman and Mr. Jonas Soenen (KU Leuven) for their support and useful suggestions. This research received funding by KU Leuven: • Research Fund (projects C16/15/059, C3/19/053, C24/18/022, C3/20/117, C3I-21-00316), Industrial Research Fund (Fellowships 13-0260, IOFm/16/004) and several Leuven Research and Development bilateral industrial projects; • Flemish Government Agencies: ○ FWO: EOS Project no G0F6718N (SeLMA), SBO project S005319N, Infrastructure project I013218N, TBM Project T001919N; PhD Grant (SB/1SA1319N), ○ EWI: the Flanders AI Research Program, ○ VLAIO: CSBO (HBC.2021.0076) Baekeland PhD (HBC.20192204) • European Commission: European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Adv. Grant grant agreement No 885682); • Other funding: Foundation 'Kom op tegen Kanker', CM (Christelijke Mutualiteit)

REFERENCES

Botman, L., Soenen, J., Theodorakos, K., Yurtman, A., Bekker, J., Vanthournout, K., Blockeel, H., Moor, B. D., and Lago, J. (2022). A scalable ensemble approach to forecast the electricity consumption of households. *IEEE Transactions on Smart Grid*.

Chengalur-Smith, I., Ballou, D., and Pazer, H. (1999). The impact of data quality information on decision making: an exploratory analysis. *IEEE Transactions on Knowledge and Data Engineering*.

Data Europa EU (2021). *Data Quality Guidelines*. Publications Office of the European Union.

Ehrlinger, L., Haunschmid, V., Palazzini, D., and Lettner, C. (2019). A daql to monitor data quality in machine learning applications. In *Prooceedings of the 30th International Conference on Database and Expert Systems Applications - Part I*.

Ehrlinger, L. and Wöß, W. (2017). Automated data quality monitoring. In *Proceedings of the 22nd MIT International Conference on Information Quality (ICIQ 2017)*.

Ehrlinger, L. and Wöß, W. (2022). A survey of data quality measurement and monitoring tools. *Frontiers in Big Data*.

Eurostat (2020). Energy statistics - supply, transformation and consumption. https://www.eea.europa.eu/data-and-maps/data/external/supply-transformation-consumption-electricity-annual-data.

Kiefer, C. (2019). Quality indicators for text data. In *BTW 2019 – Workshopband*.

Oliveira, O. and Oliveira, B. (2022). An extensible framework for data reliability assessment. In *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*.

Sadiq, S. and Indulska, M. (2017). Open data: Quality over quantity. *International Journal of Information Management*.

Soenen, J., Yurtman, A., Becker, T., D'hulst, R., Vanthournout, K., Meert, W., and Blockeel, H. (2023). Scenario generation of residential electricity consumption through sampling of historical data. *Sustainable Energy, Grids and Networks*.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific Data*.