



PIA - A Concept for a Personal Information Assistant for Data Analysis and Machine Learning of Time-Continuous Data in Industrial Applications


Christopher Schnur  ¹

Tanja Dorst  ¹

Kapil Deshmukh  ¹


Sarah Zimmer  ²

Philipp Litzenburger  ³

Tizian Schneider  ¹

Lennard Margies  ²

Rainer Müller ²

Andreas Schütze  ¹

1. Lab for Measurement Technology, Saarland University, Saarbruecken.

2. ZeMA - Center for Mechatronics and Automation Technology gGmbH, Saarbruecken.


3. Chair of Assembly Systems, Saarland University, Saarbruecken.



Date Received:

2023-02-14

Licenses:

This article is licensed under: 

Keywords:

Machine Learning, Data Analysis, Measurement and data planning

Data availability:

This publication uses no data.

Software availability:

The concept demonstrator can be found on GitHub: <https://github.com/ZeMA-gGmbH/-PIA>

Abstract. A database with high-quality data must be given to fully use the potential of Artificial Intelligence (AI). Especially in small and medium-sized companies with little experience with AI, the underlying database quality is often insufficient. This results in an increased manual effort to process the data before using AI. In this contribution, the authors developed a concept to enable inexperienced users to perform a first data analysis project with machine learning and record data with high quality. The concept comprises three modules: accessibility of (meta)data and knowledge, measurement and data planning, and data analysis. Furthermore, the concept was implemented as a front-end demonstrator on the example of an assembly station and published on the GitHub platform for potential users to test and review the concept.

1 Introduction

2 Data and their analysis play a crucial role in research and science. In recent years, especially
3 with steadily increasing computational power and the advances in Artificial Intelligence (AI),
4 the importance of high-quality data has continued to grow. In this context, entire research fields
5 and committees exclusively focus on improving data and their quality to maximize the potential
6 of their use. However, using AI in the industry also offers enormous benefits for companies.
7 In typical condition monitoring tasks, for example, early detection of damages or wear down
8 of machine parts can avoid unplanned machine downtime costs. Instead, maintenance can
9 then be scheduled, and downtimes can thus be minimized. Especially small and medium-sized

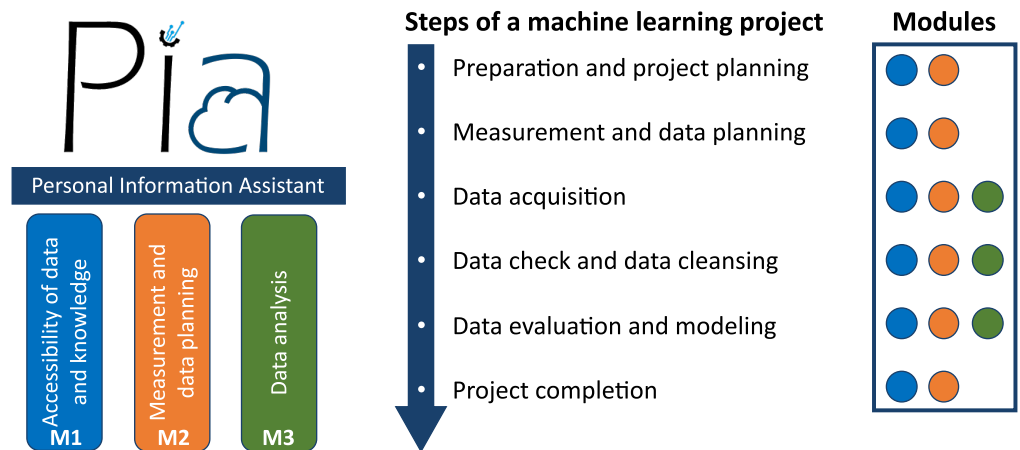


Figure 1: Concept of the Personal Information Assistant PIA, with its three modules and their contribution to the six steps of a machine learning project.

enterprises (SMEs) often have no dedicated department, skilled staff, or resources for analyzing their data and performing machine learning (ML) [1]. For these cases and to retain the obtained knowledge, the concept “PIA – Personal Information Assistant for Data Analysis” has been developed. PIA is an open-source framework based on Angular 13.3.4, a platform for building mobile and desktop web applications, which runs locally on a server and can be accessed via the intranet. PIA is developed following the in research widely accepted FAIR (Findable, Accessible, Interoperable, and Reusable) data principles and aims to transfer and apply these principles in the industry as well [2].

The concept for PIA consists of three complementing modules that support users in different stages of the machine learning project:

- **Module M1:** Accessibility of data and knowledge
- **Module M2:** Measurement and data planning
- **Module M3:** Data analysis

Figure 1 shows the three modules (M1-M3) as pillars of PIA. Furthermore, the steps of a machine learning project are shown with the involved modules in their respective color. It can be seen that M1 and M2 cover all steps of a machine learning project and are therefore closely connected to each other in the concept of PIA.

In M1 (*Accessibility of data and knowledge*), PIA provides an easy interface to access knowledge and data through the intranet. Here, two well-established methods in project management for lessons learned were combined and implemented as a knowledge base into PIA. Furthermore, an intuitive user interface (UI) enables users to easily find and access relevant (meta)data. In M2 (*Measurement and data planning*), PIA provides a checklist that was developed by Schnur et al. [3], [4] in a previous project on brownfield assembly lines to increase data quality. An English checklist version can be found in [5]. M3 (*Data analysis*) is based on the automated ML toolbox of Dorst et al. [6] and Schneider et al. [7], [8], which was developed in previous projects and successfully applied to industrial time-continuous data.

36 To the authors' knowledge, the concept PIA is a novel concept that covers a holistic approach
37 to enable inexperienced industrial users to perform a first data analysis project, focusing on
38 the domain of measurement and data planning. Here, the three modules are combined in a
39 complementing manner and brought into an interface that assists users in data analysis and
40 ensures the recordings of high-quality and "FAIR" data. Furthermore, a demonstrator for the
41 concept has been developed as a front-end in Angular 13.3.4 and tested on an assembly line as a
42 use case, which assembles a specific product in several variants, focusing on bolting processes.
43 The structure of the paper is as follows: In the next chapter, the theoretical background and
44 methodology will be explained, starting by pointing out the flaws of data in the industry, followed
45 by the three modules **M1-M3**. In the chapter *Implementation and Results*, the use case *Assembly*
46 *Line* is introduced, and details about the environment and structure of PIA are given. Thereafter,
47 the implementation and application of the concept to each module is shown. The chapter
48 *Conclusion and Outlook* summarizes the presented concept and what future research of PIA will
49 cover.

50 **2 Theoretical Background and Methodology**

51 In the chapter *Theoretical Background and Methodology* first, data in an industrial context and
52 its common problems will be elaborated on. Thereafter, each of the three modules (**M1-M3**)
53 with their respective methods will be explained.

54 **2.1 Data in Industry**

55 In their empirical study, Bauer et al. [1] found that the lack of sufficient employees (with ML
56 knowledge) and limited budget are part of the most frequent significant challenges for SMEs.
57 This can lead to rushed approaches, which end in a database with low-quality data. However, an
58 essential requirement for a successful application of AI in the industrial context is a database
59 with high-quality data, e.g., from production and testing processes. The practical application of
60 AI algorithms often fails due to

- 61 • Insufficient data quality due to missing or incomplete data annotation
- 62 • Incomplete data acquisition
- 63 • Problems linking measurement data to the corresponding manufactured products
- 64 • Lack of synchronization between different data acquisition systems

65 as shown in [9]. Furthermore, industrial data are typically acquired continuously without saving
66 relevant metadata. In addition, this often leads to a brute force approach, which tries to use
67 all acquired data. Large data sets are subsequently challenging to manage, and their use is
68 computationally expensive. A knowledge-driven approach can efficiently use resources and
69 increase the information density within the data, e.g., by reducing the amount of used sensor data
70 due to process knowledge. By recording data in a targeted manner, redundancies can also be
71 avoided. The necessary process knowledge to analyze data, especially in SMEs, is often limited
72 to a few employees and cannot be easily accessed by colleagues. Those specialists might also not

73 be willing to share their knowledge in fear they lose their distinctiveness against other employees
74 [10]. In the worst case, the (process) knowledge is lost if the specialist leaves the company.

75 2.2 Module 1 - Accessibility of Data and Knowledge

76 Due to the specific challenges in the analysis of industrial data mentioned by Wilhelm et al.
77 [9], the accessibility of the data itself and the accessibility of domain-specific knowledge play
78 a crucial role in obtaining robust ML models or further insights and knowledge about, e.g.,
79 products or processes. Module 1 of PIA, therefore, consists of a UI to access easily (meta)data
80 and knowledge as well as a knowledge base in the form of a lessons learned register. **M1** can be
81 seen as a complementing add-on component to internal knowledge repositories like company
82 wikis [11] and databases like *InfluxDB* [12] or *MongoDB* [13]. Guidance on recording and
83 structuring data and metadata can be found in the checklist [4] presented in chapter 2.3 as well.

84 To learn from previous projects and retrieve knowledge to use in future projects, lessons learned
85 are a well-established method in project management [14]. In the context of the framework PIA,
86 the lessons learned register is organized and structured like the checklist presented in **M2** and
87 contains high-quality lessons learned for each specific chapter of the checklist. Rowe et al. [14]
88 structure the formulation of lessons learned into the five subsequent steps: identify, document,
89 analyze, store, and retrieve. Moreover, they describe these steps in more detail and provide a
90 template for lessons learned. The technical standard DOE-STD-7501-99 [15] suggests that each
91 lesson learned should contain the following five elements:

- 92 • Understandable explanation of the lesson
- 93 • Context on how the lesson was learned
- 94 • Advantages of applying the lesson and potential future applications
- 95 • Contact information for further information
- 96 • Key data fields increase the findability

97 Additionally, Patton [16] distinguishes lessons learned into *lessons learned hypothesis* and
98 *high-quality lessons learned*. While a *lessons learned hypothesis* is a lesson learned with one
99 supporting evidence, *high-quality lessons learned* could be approved in multiple projects. To
100 ensure the quality of the lessons learned, Patton [16] further formulated ten questions in his
101 paper for generating such high-quality lessons learned. Moreover, he recommended reviewing
102 lessons learned periodically regarding their usefulness and sorting out obsolete lessons learned
103 to maintain high quality.

104 Figure 2 shows the approach proposed in this contribution. After analyzing a given use case,
105 specific results were achieved. The whole project is evaluated in a retrospective, and lessons
106 learned are formulated according to the five steps of Rowe et al. [14]. If the lessons learned
107 (hypothesis) can be validated in further projects, they get added to the lessons learned register.
108 The lessons learned register is reviewed regularly to ensure relevance and actuality.

Formulation of High-Quality Lessons Learned

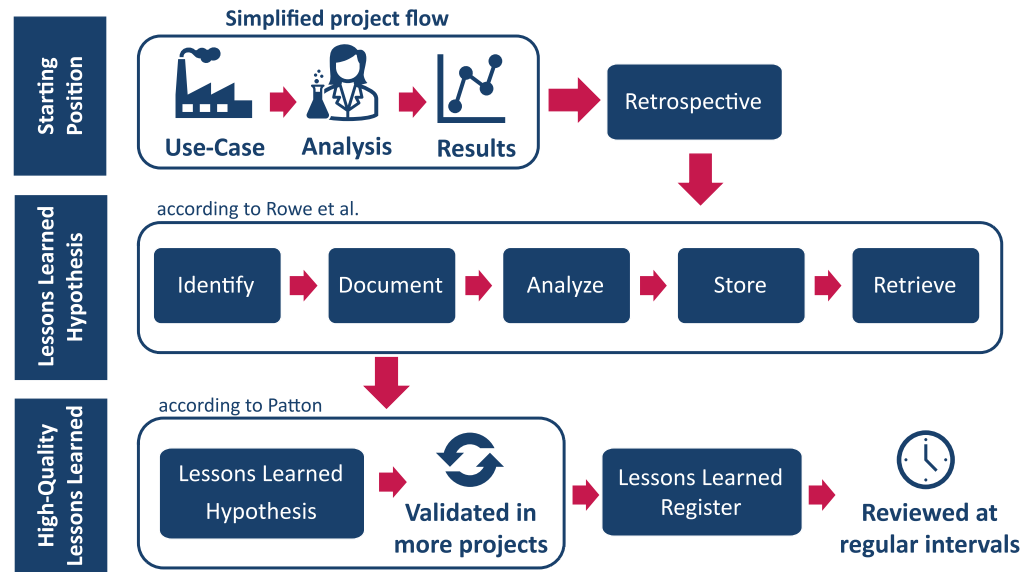


Figure 2: Formulation of high-quality lessons learned through combining the approaches of Rowe et al. and Patton [14], [16].

109 2.3 Module 2 - Checklist for Measurement and Data Planning

110 The foundation for successfully applying AI in industry is a high quality of the underlying data.
 111 The research field measurement and data planning can be seen as an early part or requirement of
 112 data mining, which is the process of extracting knowledge from data sets using computational
 113 techniques [17]. For industrial data mining, the *Cross-Industry Standard Process for Data*
 114 *Mining* (CRISP-DM) established, which divides data mining into the six non-sequential and
 115 independent phases: business understanding, data understanding, data preparation, modeling,
 116 evaluation, and deployment [18]. Since data mining consists of several disciplines, each of
 117 which has its own research area, inexperienced users can feel overwhelmed and demotivated
 118 mainly because the industry's focus is not to record high-quality data but to use data to increase
 119 efficiency and margin. A guide or checklist can help users to orientate and access the field of data
 120 mining. Here several approaches can be found in literature like, e.g., *A Checklist for Analyzing*
 121 *Data* of [19] or the *Analytical Checklist – A Data Scientist's Guide for Data Analysis* [20]. The
 122 checklists mentioned are universally applicable but lack information for the realization of a data
 123 analysis project and assume that data sets are already recorded. Other concepts, like, e.g., *FAIR*
 124 of Wilkinson et al. [2] ensure high-quality data and offer practicable solutions like *The FAIR*
 125 *Cookbook* that guide new users but are primarily focused on research data.

126 The *Checklist - Measurement and data planning for machine learning in assembly* of Schnur
 127 et al. [3] tries to find the sweet spot between having a broad scope, transferring knowledge
 128 from research data management but still being clearly structured and not overwhelming for an
 129 inexperienced user. Within PIA, the checklist enables the users of PIA to perform a machine
 130 learning project from the beginning to the end and record high-quality FAIR data. It covers the
 131 following chapters:

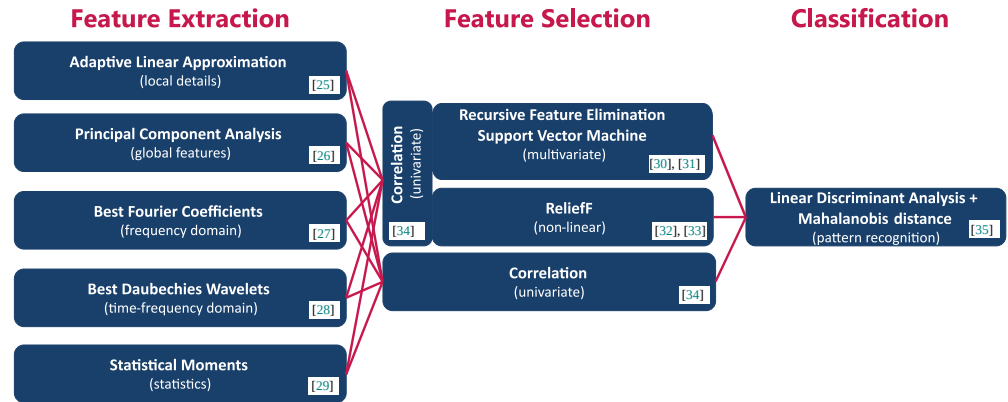


Figure 3: Algorithms of the automated ML toolbox for classification with their corresponding literature (adapted from [6]).

- 132 • Preparation and project planning
- 133 • Measurement and data planning
- 134 • Data acquisition
- 135 • Data check and data cleansing
- 136 • Data evaluation and modeling
- 137 • Project completion

138 Each chapter begins with a short introduction, followed by checkpoints that guide the user. Here,
 139 two types of checkpoints exist: necessary and best-practice checkpoints. While the best-practice
 140 checkpoints are optional but highly recommended, the necessary checkpoints must be executed.
 141 Furthermore, the checklists provide tips and notes derived from high-quality lessons learned
 142 from previous ML projects and further literature suggestions. The checklist is based on a revised
 143 version of the CRISP-DM mentioned above. Therefore, some parts of the checklist are iterative.
 144 The checklist was initially published in German on the file-sharing platform Zenodo and has been
 145 translated for integration into PIA to English, which also increases accessibility and re-usability
 146 [4].

147 2.4 Module 3 - Data Analysis

148 In nowadays industry, an extensive range of tools and software for data analysis exists. Some
 149 well-known and established examples are the software library *Pandas* for Python [21], Power
 150 BI [22], the *Statistics and Machine Learning Toolbox* of MATLAB[®] [23] or the open platform
 151 *KNIME* [24]. All the solutions mentioned are powerful data analysis tools and offer a broad
 152 spectrum of algorithms.

153 Within the concept of PIA, the focus lies on time-continuous data. However, potential users can
 154 implement every ML tool or algorithm that suits their use case and data into **M3**, starting from
 155 traditional approaches like feature extraction, selection, and classification/regression to modern

156 approaches like deep learning with neural networks. However, it could be shown by Goodarzi
157 et al. [36] that traditional approaches could perform similar to modern approaches while being
158 less complex and have a higher interpretability. To cope with the heterogeneous data sources in
159 industrial applications, especially as an inexperienced user, a set of different feature extraction
160 and selection methods can be beneficial [37]. For the implementation of PIA within this study,
161 the authors use the existing automated ML toolbox for time-continuous data of Dorst et al. [6] and
162 Schneider et al. [7]. This toolbox automatically tests different combinations of feature extraction
163 and feature selection methods with linear discriminant analysis and Mahalanobis distance as the
164 classifier. This automated ML toolbox combines five complementary feature extraction methods
165 with three feature selection methods, as shown with their corresponding literature in Figure 3.
166 A 10-fold cross-validation automatically determines the best of the resulting 15 combinations
167 by ranking the combinations according to their resulting cross-validation error on the test data
168 [38], [39]. Due to the different focus of each algorithm (shown in Figure 3), the toolbox could
169 achieve good results in a broad application range ([37], [40]).

170 Users can perform a first ML analysis with the toolbox by running five lines of code:

```
171  
172 1 addPaths; %Adds folders and subfolders to the path  
173 2 load dataset.mat %Load data set  
174 3 fulltoolbox = Factory.FullToolboxMultisens(); %Build object  
175 4 fulltoolbox.train(data, target); %Train model with data and target as  
176   input  
177 5 prediction = fulltoolbox.apply(data); %Apply trained model on data  
178
```

Listing 1: Code to run the complete toolbox.

179 For further analysis or regression tasks, the methods can be modified, changed, or applied
180 separately.

181 3 Implementation and Results

182 To evaluate the methods presented in the chapter *Theoretical Background and Methodology*,
183 the use case *Assembly Line* will be first introduced in this chapter. After that, PIA's chosen
184 environment and structure will be shown, followed by the implementation of the three modules
185 regarding the given use case.

186 3.1 Use Case: Assembly Line

187 As a validation use case for this contribution, an assembly line with two stations was chosen
188 (Figure 4 a) that produces a device holder (Figure 4 b). In the first station, a robot picks up
189 the individual parts of the device holder from a warehouse and places them on a workpiece
190 carrier. The product is transported to station 2 by a belt conveyor for the next step. There, a
191 worker assembles the two components by a bolting process. In addition, the device holder can
192 be produced in another variant (Figure 4 c). The use case is presented in more detail in [41].

193 The combination of two different stations with different processes and different degrees of
194 automation, as well as the opportunity to produce a second variant of the device holder, make

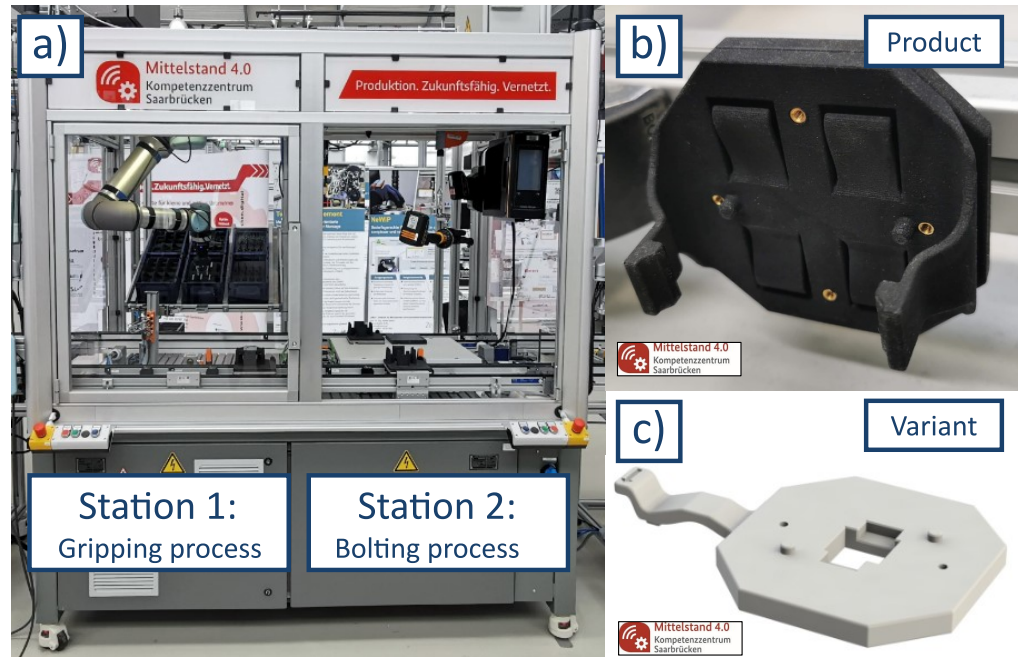


Figure 4: Picture of the assembly line with its two stations (a), the produced device holder (b) and a variant of the device holder (c).

195 this assembly line a good use case for demonstrating the flexibility of PIA while keeping the
 196 complexity low (compared to more extensive assembly lines).

197 3.2 Environment and Structure of PIA

198 The Angular framework, an open-source single-page web application framework, has been
 199 chosen to demonstrate the concept of PIA. Angular 13.3.4. allows fast development of the
 200 demonstrator and gives the possibility that, once the demonstrator is hosted on the web, PIA can
 201 be easily accessed from any device within the intranet.

202 Figure 5 shows a schematic representation of the development environment of PIA. For simulating
 203 the user experience, the type-2 hypervisor Oracle VM VirtualBox (<https://www.virtualbox.org/>)
 204 from Oracle Corporation was used with Ubuntu as the guest operating system (OS).
 205 However, in general, Angular can also be used on Microsoft Windows or Apple macOS.

206 Table 1 gives an overview of the used environments and Table 2 of the used libraries, each with
 207 their corresponding sources.

Value	Package	Source
E1	Angular CLI	https://www.angular.io/
E2	Node JS	https://www.nodejs.org/en/
E3	Node Package Manager	https://www.npmjs.com/

Table 1: Overview of the used environments.

208 Besides Angular (E1), the environments Node JS (E2) and Node Package Manager (E3) are used.
 209 Angular's primary architectural features are a hierarchy of components. Using this structure,

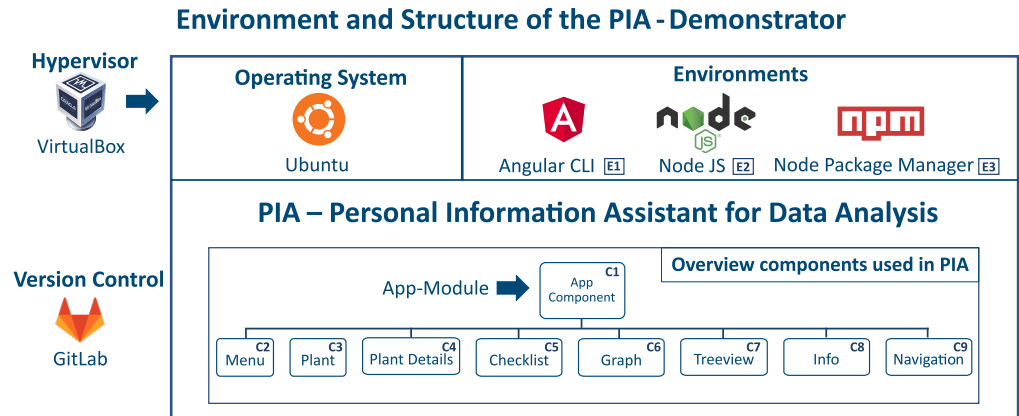


Figure 5: Schematic representation of the environment and structure of the PIA-Demonstrator.

Value	Package	Source
L1	Angular Forms	https://www.npmjs.com/package/@angular/forms
L2	Angular Material	https://www.material.angular.io/
L3	Bootstrap	https://www.npmjs.com/package/bootstrap
L4	Charts js	https://www.npmjs.com/package/chart.js
L5	Flex Layout	https://www.npmjs.com/package/flex-layout

Table 2: Overview of the used libraries.

210 the various PIA functionalities have been separated into components for ease of use and reuse.
 211 Table 3 provides an overview with a short description of the components used in PIA. This
 212 structure also allows to easily add new components to the application without interfering with
 213 existing ones. The Angular Material library (L2) provides a consistent experience across the
 214 website. Specific dynamic components have also been made responsive using Bootstrap and
 215 Flex-Layout libraries (L3, L4). To make it easier for future developers to add new information to
 216 the website, data about each process has been saved in JSON format and then queried to display
 217 the relevant information in the UI. Users or developers can easily add more plants or tools to the
 218 application by editing the relevant JSON file, which will be dynamically displayed in the UI.

219 Figure 6 shows the landing page of PIA. Over a menu, the user can navigate through the four
 220 menu points:

- 221 1. Plant
- 222 2. Knowledge base
- 223 3. Checklist
- 224 4. Data Analysis

225 3.3 Module 1 - Accessibility of Data and Knowledge

226 The implementation of **M1** contains two parts, accessibility of data and metadata (menu-point:
 227 *Plant*) and a lessons learned register (*Knowledge base*). The plant module displays information,
 228 data, and metadata about various plants. Figure 7 shows an example flow-through of the use case

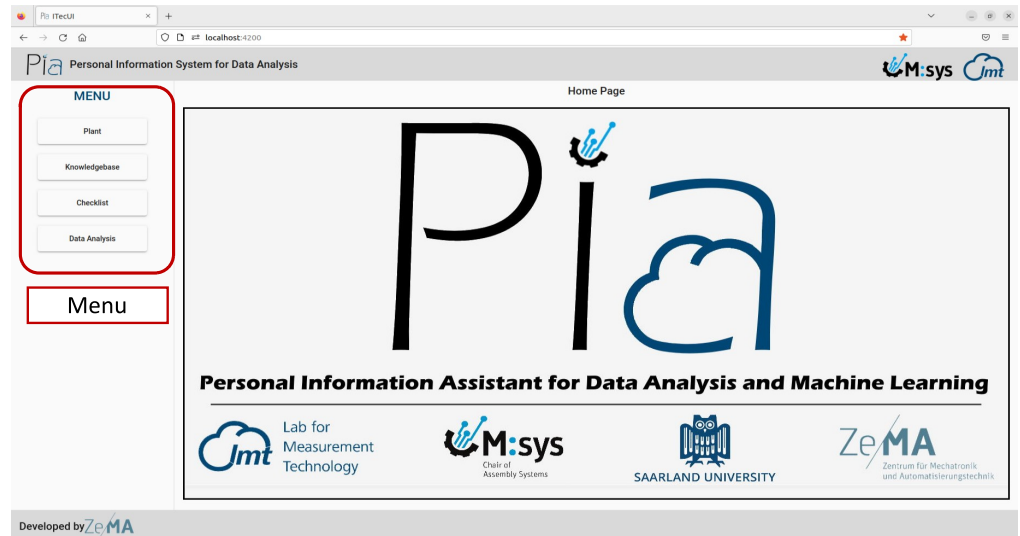


Figure 6: Landing page of PIA with its four menu points (red box): Plant, Knowledge base, Checklist, and Data Analysis.

229 in this study. After clicking on the *Plant* button, the available stations of the plant are displayed:
 230 *Gripping Process* and *Bolting Process*. After selecting a process (Figure 7, green box), the user
 231 can select between the following options:

- 232 • **Product:** Displays all available products with their variants containing further information
 233 like CAD files and technical drawings.
- 234 • **Resources:** All process resources are displayed with a picture (Figure 7, red box) and
 235 contain sub menus (Figure 7, blue box) with further information.
- 236 • **Measurements:** Data can be loaded as a CSV-file into PIA and plotted through the *charts.js*
 237 (*L4*) library.
- 238 • **Video:** A video of the process that shows the procedure and allows the user to develop a
 239 better understanding and link the data of a process. The video was embedded using the
 240 *HTML iframe tag*.
- 241 • **Sensors:** Contains an overview of all used sensors and their metadata (like sensor type,
 242 sensor position, sampling rate, etc.).
- 243 • **Shift book:** Displays the digital version of the shift book. Using the entries of the shift
 244 book can support the user, e.g., to explain outliers or shifts in data.

245 All information is contained in an array of JavaScript objects. Therefore, a new plant, station,
 246 or resource can be easily included by adding new objects to the array in the same format and
 247 assigning it on the front-end. Here, Angular material cards (*L2*) are used to display further
 248 information, e.g., process resources. An example of the basic structure of the array of objects for
 249 a *Station* with one process which includes a robot and relevant metadata, e.g., technical data or
 250 technical drawings, is shown in list. 2 (Section A).

251 Further instances of the resources, e.g., a gripper for the robot, can be easily added by creating a
 252 new object with ID, name, and paths corresponding to the documents and images in the assets

Nr	Component Name	Description
C1	App Component	The application's root component is defined in the <i>app.module.ts</i> file and bootstrapped to the <i>main.ts</i> file to start the application. It acts as a container for all other components in the application.
C2	Menu	It provides a menu in the application to navigate through the various features. It appears on the left-hand side of the UI and has buttons for navigation through components.
C3	Plant	Implements the navigation to select the specific plant described in the application and provides buttons to navigate through the various embedded components.
C4	Plant Details	Implements the information about a specific plant and contains an array of objects, which saves information about the specific plant. Each object in the array contains properties that describe the plant. The main array of the plant has further arrays embedded inside, with similar properties describing the processes/stations inside a plant.
C5	Checklist	Implements the checklist with a navigation pane to move to different nodes inside the checklist. It has a JSON implementation that contains the description and other relevant information about each node in the checklist.
C6	Graph	Implements the plotting of graphs in the application with the Charts js library. It allows users to plot data from an uploaded CSV-file.
C7	Treeview	Implements the tree view of available or used processes. Furthermore, it implements the domain-specific knowledge of those processes or related tools in so-called cards.
C8	Info	Implements a card that displays specific text information regarding a particular process in the plant.
C9	Navigation	Header component, which implements the logo and name of the application

Table 3: Overview of the used components.

253 folder and providing the relative paths to the corresponding documents. The new instance will
 254 automatically be displayed in the UI after recompiling. Furthermore, the button *Knowledge base*
 255 (Figure 7, blue box) contains specific knowledge about each resource.

256 The second part of the knowledge base contains the lessons learned register and a simple example
 257 of the link to general knowledge. The general knowledge was implemented illustratively as
 258 a graphical representation of the assembly processes in the form of a tree. Here, the user can
 259 expand the tree by selecting the respective nodes to access the sub-nodes that describe the next
 260 steps of the process described in the parent node. The information component has been integrated
 261 with the nodes, which can provide further descriptive information about each node.

262 The implementation of the lessons learned register is shown in Figure 8. In the suggested version
 263 of a lessons learned register, each lesson learned is generated by the process shown in Figure 2

264 and grouped by their respective project step (chapter) of the checklist (Figure 8, blue box). After
 265 selecting a chapter, the lessons learned appear on the right-hand side (Figure 8, red box). Users
 266 can add criticism to existing lessons learned, lessons learned hypotheses, or additional files in the
 267 *Comment Section* (Figure 8, green box). The *Comment Section* is reviewed at regular intervals
 268 and, if necessary, transferred to the *Lessons Learned Register*.

269 Using the presented structured, high-quality lessons learned register allows the user to easily
 270 access domain-specific knowledge in a target-orientated manner and to avoid previously made
 271 mistakes in earlier projects. The presented GUI enables users to find relevant information
 272 intuitively rather than searching for information in multi-layered folder structures with restricted
 273 accessibility.

274 3.4 Module 2 - Checklist for Measurement and Data Planning

275 The checklist implementation uses the tree component of the Angular Material (L2) library,
 276 which allows the present hierarchical content as an expandable tree (Figure 9, blue box). Each
 277 node of this tree displays information about itself and additional tips or hints (Figure 9, red box).
 278 Furthermore, comments and files can be added to each checkpoint (Figure 9, green box). This
 279 helps employees who are new to the project to catch up and comprehend past steps. When the
 280 user ticks through all the sub-nodes, the primary process node is automatically ticked, indicating
 281 that all the sub-processes have been completed.

282 The *Checklist for Measurement and Data Planning* guides and supports users holistically through
 283 a data analysis project while highlighting trip points. Using the checklist in an implemented
 284 version rather than the printed version, the status and progress of ongoing projects can be tracked
 285 and understood by non-involved users in case a worker gets sick or leaves the company.

286 3.5 Module 3 - Data Analysis

287 Since PIA is implemented as a front-end demonstrator with no back-end, the data analysis is
 288 carried out in MATLAB[®] Online[™]. Figure 10 shows the results of the data analysis with the ML
 289 toolbox for data of the example use case provided in [6]. The toolbox can be directly connected
 290 to GitHub into MATLAB[®] Online[™] (Figure 10). As shown in the blue box of Figure 10, the

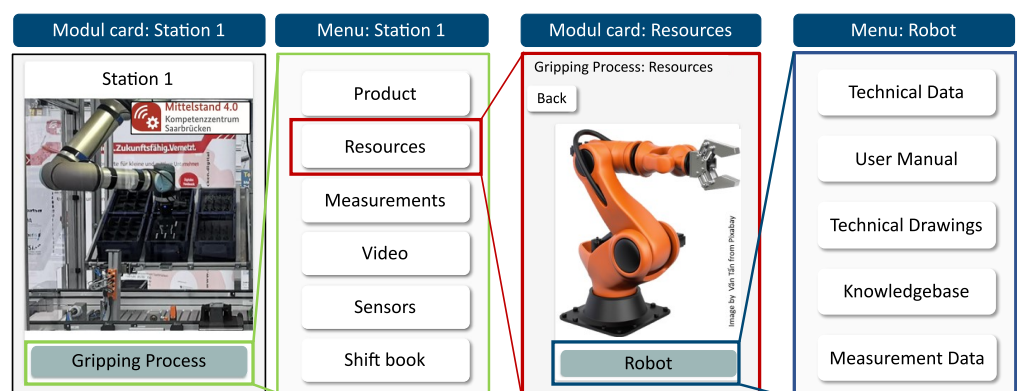


Figure 7: Schematic representation of the knowledge base with its single components.

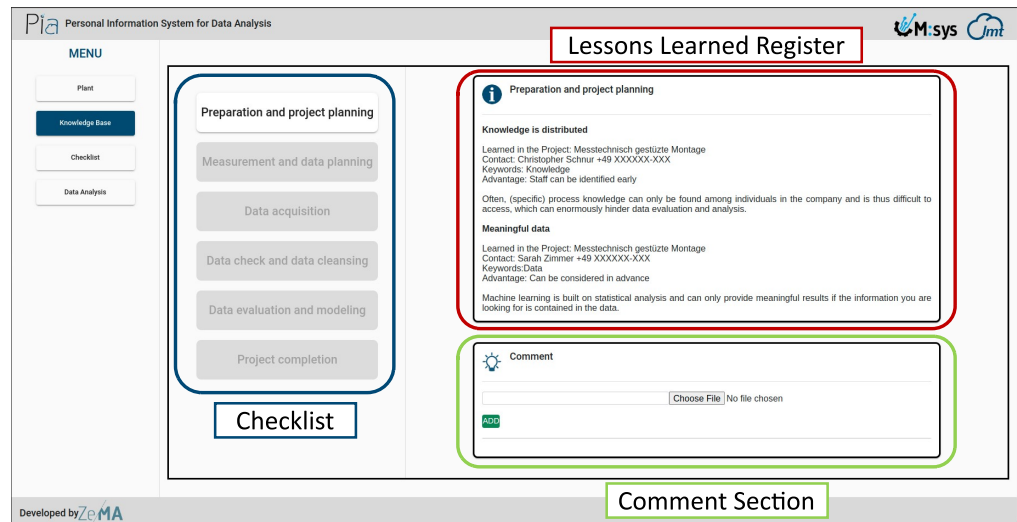


Figure 8: Implementation of the lessons learned register in PIA. Blue box: Chapters of the checklist. Red box: Lessons learned register. Green box: Comment section.

291 user can access other algorithms by clicking through the folder structure. After executing the
 292 code (Figure 10, green box), the user can plot the results (Figure 10, red box). For interpreting
 293 the results, the user can follow the subsequent steps of the checklist in **M2** while using the
 294 knowledge, data, and metadata provided in **M1**.

295 The implemented toolbox can be used by users with little programming experience while covering
 296 a broad range of algorithms for analyzing industrial time-continuous data with ML. However,
 297 the users can also embed other, e.g., use-case-specific algorithms.

298 4 Conclusion and Outlook

299 The personal information assistant PIA supports inexperienced users in performing an ML project
 300 and gaining further insights from data. For this, it consists of the three modules *Accessibility*
 301 *of Data and Knowledge*, *Checklist for Measurement and Data Planning*, and *Data Analysis*.
 302 *Accessibility of Data and Knowledge* allows the user to access relevant metadata and gain
 303 knowledge about the plant and processes through a lessons learned register. Using the concept,
 304 data, and metadata can be recorded and organized in a targeted and structured manner, creating
 305 appropriate boundary conditions for data analysis projects. In its current version, the PIA
 306 demonstrator is implemented on a front-end in a virtual machine. By adding an additional
 307 back-end, the progress of a project, attached files, comments, etc., can be saved and loaded
 308 properly. Furthermore, a direct connection to a database is conceivable. Implementing PIA
 309 in Angular was a time-efficient way to demonstrate its benefits. However, users can decide
 310 if they want to apply the concept in a different framework. Furthermore, the modules can be
 311 switched or customized to the specific needs of the users due to the open-source nature of this
 312 contribution and the PIA concept in general. The authors will further develop their concept in
 313 future research and test it on other use cases. Current development focuses on the integration of
 314 ontology's, respectively machine-readable metadata in **M1**, the generalization of the checklist in
 315 **M2**, and improvements regarding the usability for inexperienced users in SMEs and, in **M3**, the

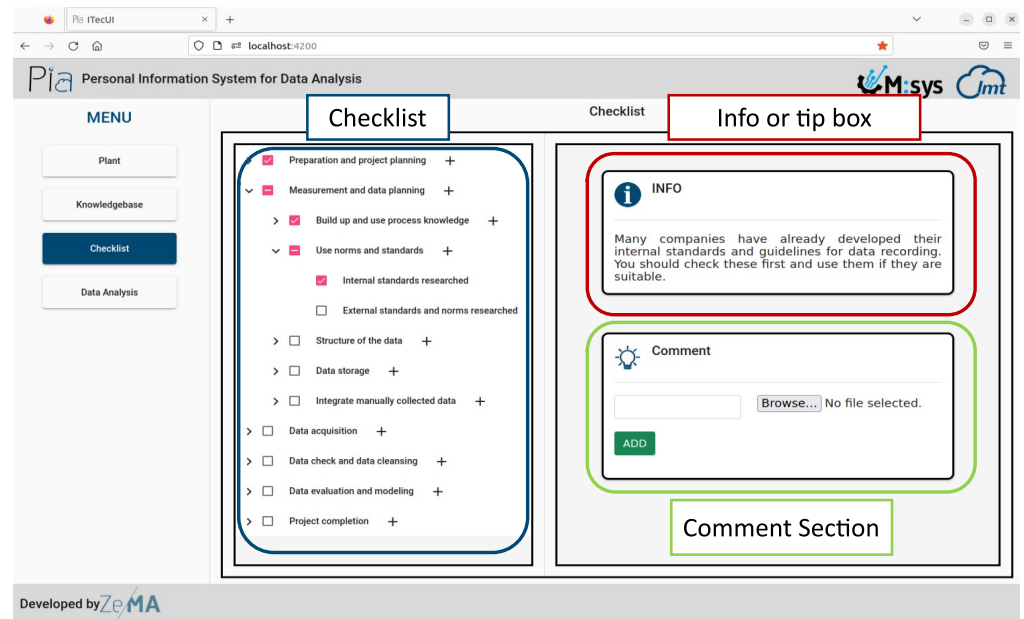


Figure 9: Implementation of the checklist in PIA. Blue box: Chapters of the checklist with their corresponding sub-chapters and checkboxes. Red box: Info- and Tip-boxes. Green box: The comment section.

- 316 integration of a data pipeline to evaluate the data quality as well as the usage of algorithms that
 317 consider measurement uncertainty.

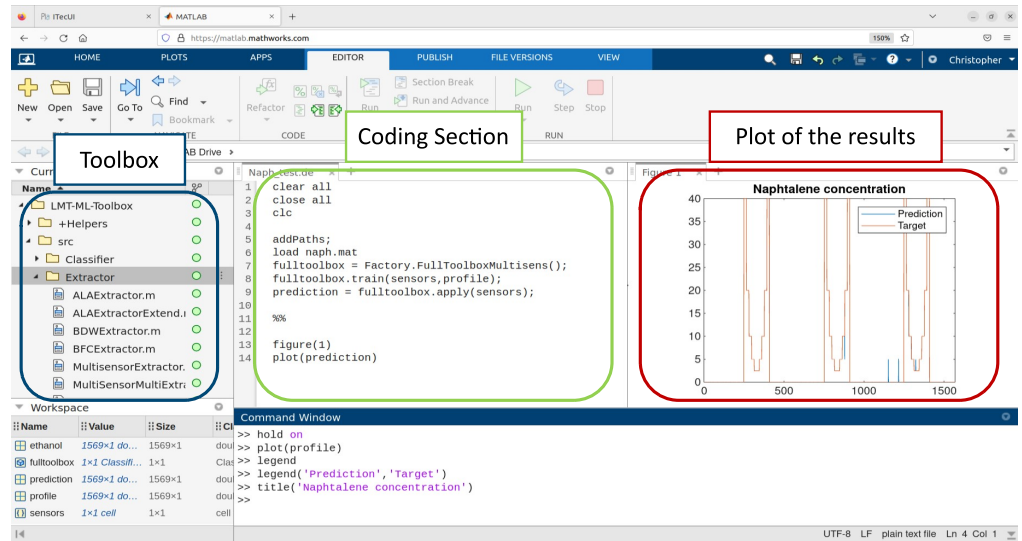


Figure 10: Screenshot of the automated ML toolbox connected via GitHub in MATLAB[®] Online. Blue box: Folder structure of the toolbox. Green box: Coding section. Red box: Plot of the results.

318 A Code sample

```

319
320 1 const Station = [
321 2   {
322 3     id: '1',
323 4     title: 'Station 1',
324 5     img: "path/station_1/picture_station_1.jpg",
325 6     process_1: 'Process XY',
326 7     resource_1:[
327 8       {id: '1',
328 9         name: 'Robot XY',
329 10        technicalName: 'process 1 technical name',
330 11        img:"path/station_1/Process_1/Robot_XY/Robot_XY.jpg",
331 12        technicalData:"path/station_1/Process_1/Robot_XY/DataSheets/
332 13        Robot_technical_details.pdf",
333 14        manual:"path/station_1/Process_1/Robot_XY/DataSheets/DataSheets
334 15        /Robot_manual.pdf",
335 16        technicalDrawing:"path/station_1/Process_1/Robot_XY/DataSheets/
336 17        Robot_technical_drawing.pdf"
337 18      }
338 19    ]
339 20  }
340 21  ]

```

Listing 2: Sample code for a station.

340 5 Acknowledgements

341 This work was funded by the European Regional Development Fund (ERDF) in the framework
 342 of the research projects within the framework of the research projects "Messtechnisch gestützte
 343 Montage" and "iTecPro – Erforschung und Entwicklung von innovativen Prozessen und Tech-
 344 nologien für die Produktion der Zukunft". Future development is carried out in the project
 345 "NFDI4Ing – the National Research Data Infrastructure for Engineering Sciences", funded by
 346 the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 442146713.

347 Furthermore, the authors thank Anne Blum and Dr.-Ing. Leonie Mende for their profound
 348 input during conceptualization and the Mittelstand 4.0-Kompetenzzentrum Saarbrücken for the
 349 deployment of a use case.

350 6 Roles and Contributions

351 **Christopher Schnur:** Conceptualization, Writing & original draft

352 **Tanja Dorst:** Conceptualization, review & editing

353 **Kapil Deshmukh:** Programming & implementation

354 **Sarah Zimmer:** Conceptualization

355 **Philipp Litzenburger:** Conceptualization

356 **Tizian Schneider:** Methodology & review

357 **Lennard Margies:** Coordination

358 **Rainer Müller:** Concept & Coordination

359 **Andreas Schütze:** Coordination, Concept & review

360 References

- 361 [1] M. Bauer, C. van Dinther, and D. Kiefer, "Machine learning in sme: An empirical study
 362 on enablers and success factors," in *AMCIS 2020 Proceedings*, 2020.
- 363 [2] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, *et al.*, "The FAIR Guiding Principles
 364 for scientific data management and stewardship," *Scientific Data*, vol. 3, no. 1, p. 160 018,
 365 2016. DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18).
- 366 [3] C. Schnur, S. Klein, A. Blum, A. Schütze, and T. Schneider, "Steigerung der Datenqualität
 367 in der Montage," *wt Werkstattstechnik online*, vol. 112, pp. 783–787, Dec. 2022. DOI:
 368 [10.37544/1436-4980-2022-11-12-57](https://doi.org/10.37544/1436-4980-2022-11-12-57).
- 369 [4] C. Schnur, S. Klein, and A. Blum, *Checkliste – Mess- und Datenplanung für das maschinelle*
 370 *Lernen in der Montage*, version 7, Aug. 2022. DOI: [10.5281/zenodo.6943476](https://doi.org/10.5281/zenodo.6943476).
- 371 [5] C. Schnur, S. Klein, and A. Blum, *Checklist – Measurement and data planning for machine*
 372 *learning in assembly*, version 7, Jan. 2023. DOI: [10.5281/zenodo.7556876](https://doi.org/10.5281/zenodo.7556876).

- 373 [6] T. Dorst, Y. Robin, T. Schneider, and A. Schütze, “Automated ML Toolbox for Cyclic
374 Sensor Data,” in *MSMM 2021 - Mathematical and Statistical Methods for Metrology*,
375 (Online, May 31–Jun. 1, 2021), 2021, pp. 149–150. [Online]. Available: http://www.msmm2021.polito.it/content/download/245/1127/file/MSMM2021_Booklet_c.pdf
376 (visited on 01/24/2023).
377
- 378 [7] T. Schneider, N. Helwig, and A. Schütze, “Industrial condition monitoring with smart
379 sensors using automated feature extraction and selection,” *Measurement Science and*
380 *Technology*, vol. 29, no. 9, 2018. DOI: [10.1088/1361-6501/aad1d4](https://doi.org/10.1088/1361-6501/aad1d4).
- 381 [8] T. Schneider, N. Helwig, and A. Schütze, “Automatic feature extraction and selection
382 for classification of cyclical time series data,” *tm - Technisches Messen*, vol. 84, no. 3,
383 pp. 198–206, 2017. DOI: [10.1515/teme-2016-0072](https://doi.org/10.1515/teme-2016-0072).
- 384 [9] V. Gudivada, A. Apon, and J. Ding, “Data Quality Considerations for Big Data and
385 Machine Learning: Going Beyond Data Cleaning and Transformations,” *International*
386 *Journal on Advances in Software*, vol. 10, pp. 1–20, Jul. 2017.
- 387 [10] S. Wang and R. A. Noe, “Knowledge sharing: A review and directions for future research,”
388 *Human Resource Management Review*, vol. 20, pp. 115–131, 2010. DOI: [10.1016/j.hr](https://doi.org/10.1016/j.hr)
389 [mr.2009.10.001](https://doi.org/10.1016/j.hr).
- 390 [11] A. Majchrzak, C. Wagner, and D. Yates, “Corporate Wiki Users: Results of a Survey,” in
391 *Proceedings of the 2006 International Symposium on Wikis*, (Odense, Denmark, Aug. 21–
392 23, 2006), New York, NY, USA: Association for Computing Machinery, 2006, pp. 99–104.
393 DOI: [10.1145/1149453.1149472](https://doi.org/10.1145/1149453.1149472).
- 394 [12] InfluxData, *Influxdb. it’s about time*. 2013. [Online]. Available: <https://www.influxdata.com/influxdb/>.
395
- 396 [13] K. Banker, *MongoDB in Action*. USA: Manning Publications Co., 2011, ISBN: 1935182870.
- 397 [14] S. F. Rowe and S. Sikes, “Lessons learned: Taking it to the next level,” in *PMI® Global*
398 *Congress*, (Seattle, WA, USA, Oct. 21–24, 2006), 2006.
- 399 [15] US Department of Energy, *The DOE Corporate Lessons Learned Program*, DOE-STD-
400 7501-99, Dec. 1999. [Online]. Available: <https://www.standards.doe.gov/standards-documents/7000/7501-astd-1999/@images/file> (visited on 01/24/2023).
401
- 402 [16] M. Q. Patton, “Evaluation, Knowledge Management, Best Practices, and High Quality
403 Lessons Learned,” *American Journal of Evaluation*, vol. 22, no. 3, pp. 329–336, Sep.
404 2001. DOI: [10.1177/109821400102200307](https://doi.org/10.1177/109821400102200307).
- 405 [17] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann,
406 2022.
- 407 [18] P. Chapman, J. Clinton, R. Kerber, *et al.*, *CRISP-DM 1.0: Step-by-step data mining guide*,
408 2000. [Online]. Available: <https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf> (visited on 01/24/2023).
409
- 410 [19] K. L. Sainani, “A checklist for analyzing data,” *PMR*, vol. 10, no. 9, pp. 963–965, 2018,
411 ISSN: 1934-1482. DOI: <https://doi.org/10.1016/j.pmrj.2018.07.015>. [Online].
412 Available: [https://www.sciencedirect.com/science/article/pii/S19341482](https://www.sciencedirect.com/science/article/pii/S1934148218304258)
413 [18304258](https://www.sciencedirect.com/science/article/pii/S1934148218304258).

- 414 [20] S. Kangralkar, *Analytical Checklist — A Data Scientist’s Guide for Data Analysis*, 2021.
415 [Online]. Available: [https://medium.com/swlh/analytical-checklist-a-data-](https://medium.com/swlh/analytical-checklist-a-data-scientist-guide-for-data-analysis-972ed3ff1d59)
416 [scientist-guide-for-data-analysis-972ed3ff1d59](https://medium.com/swlh/analytical-checklist-a-data-scientist-guide-for-data-analysis-972ed3ff1d59) (visited on 08/11/2023).
- 417 [21] T. pandas development team, *Pandas-dev/pandas: Pandas*, version v2.0.3, If you use this
418 software, please cite it as below., Jun. 2023. DOI: [10.5281/zenodo.8092754](https://doi.org/10.5281/zenodo.8092754). [Online].
419 Available: <https://doi.org/10.5281/zenodo.8092754>.
- 420 [22] *Creating machine learning models in power bi*, Accessed: 2023-08-10, 2019. [Online].
421 Available: [https://powerbi.microsoft.com/de-de/blog/creating-machine-l](https://powerbi.microsoft.com/de-de/blog/creating-machine-learning-models-in-power-bi/)
422 [earning-models-in-power-bi/](https://powerbi.microsoft.com/de-de/blog/creating-machine-learning-models-in-power-bi/).
- 423 [23] T. M. Inc., *Statistics and machine learning toolbox version: 12.5 (r2023a)*, Natick, Mas-
424 sachusetts, United States, 2023. [Online]. Available: <https://www.mathworks.com>.
- 425 [24] M. R. Berthold, N. Cebron, F. Dill, *et al.*, “Knime - the konstanz information miner:
426 Version 2.0 and beyond,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 26–31, 2009. DOI:
427 [10.1145/1656274.1656280](https://doi.org/10.1145/1656274.1656280).
- 428 [25] R. T. Olszewski, “Generalized feature extraction for structural pattern recognition in
429 time-series data,” Ph.D. dissertation, Carnegie Mellon University, 2001, ISBN: 978-0-
430 493-53871-6.
- 431 [26] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and*
432 *Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, Aug. 1987. DOI: [10.1016/01](https://doi.org/10.1016/0169-7439(87)80084-9)
433 [69-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).
- 434 [27] F. Mörchen, “Time series feature extraction for data mining using DWT and DFT,”
435 *Department of Mathematics and Computer Science, University of Marburg, Germany -*
436 *Technical Report*, vol. 33, 2003. [Online]. Available: [https://www.mybytes.de/pape](https://www.mybytes.de/papers/moerchen03time.pdf)
437 [rs/moerchen03time.pdf](https://www.mybytes.de/papers/moerchen03time.pdf) (visited on 01/24/2023).
- 438 [28] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics,
439 1992. DOI: [10.1137/1.9781611970104](https://doi.org/10.1137/1.9781611970104).
- 440 [29] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*,
441 4th ed. Boston: McGraw-Hill, 2002, ISBN: 978-0-07-366011-0.
- 442 [30] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of*
443 *Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.
- 444 [31] A. Rakotomamonjy, “Variable selection using svm-based criteria,” *Journal of Machine*
445 *Learning Research*, vol. 3, pp. 1357–1370, Mar. 2003. DOI: [10.1162/1532443033227](https://doi.org/10.1162/153244303322753706)
446 [53706](https://doi.org/10.1162/153244303322753706).
- 447 [32] M. Robnik-Šikonja and I. Kononenko, “Theoretical and Empirical Analysis of ReliefF
448 and RReliefF,” *Machine Learning*, vol. 53, no. 1, pp. 23–69, Oct. 2003. DOI: [10.1023](https://doi.org/10.1023/A:1025667309714)
449 [/A:1025667309714](https://doi.org/10.1023/A:1025667309714).
- 450 [33] I. Kononenko and S. J. Hong, “Attribute selection for modelling,” *Future Generation*
451 *Computer Systems*, vol. 13, no. 2-3, pp. 181–195, Nov. 1997, ISSN: 0167739X. DOI:
452 [10.1016/S0167-739X\(97\)81974-7](https://doi.org/10.1016/S0167-739X(97)81974-7).

- 453 [34] J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson correlation coefficient,” in *Noise*
454 *Reduction in Speech Processing*. Berlin, Heidelberg: Springer, 2009, pp. 1–4. DOI: [10.1](https://doi.org/10.1007/978-3-642-00296-0_5)
455 [007/978-3-642-00296-0_5](https://doi.org/10.1007/978-3-642-00296-0_5).
- 456 [35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: John
457 Wiley & Sons, 2001, ISBN: 978-0-471-05669-0.
- 458 [36] P. Goodarzi, A. Schütze, and T. Schneider, *tm - Technisches Messen*, vol. 89, no. 4, pp. 224–
459 239, 2022. DOI: [doi:10.1515/teme-2021-0129](https://doi.org/10.1515/teme-2021-0129). [Online]. Available: [https://doi](https://doi.org/10.1515/teme-2021-0129)
460 [.org/10.1515/teme-2021-0129](https://doi.org/10.1515/teme-2021-0129).
- 461 [37] T. Schneider, N. Helwig, and A. Schütze, “Automatic feature extraction and selection for
462 condition monitoring and related datasets,” in *2018 IEEE International Instrumentation*
463 *and Measurement Technology Conference (I2MTC)*, 2018, pp. 1–6. DOI: [10.1109/I2](https://doi.org/10.1109/I2MTC.2018.8409763)
464 [MTC.2018.8409763](https://doi.org/10.1109/I2MTC.2018.8409763).
- 465 [38] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model
466 selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelli-*
467 *gence - Volume 2*, (Montreal, Quebec, Canada, Aug. 20–25, 1995), ser. IJCAI’95, San
468 Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143, ISBN:
469 978-1-55860-363-9.
- 470 [39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data*
471 *Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer New York, 2009,
472 ISBN: 978-0-387-84858-7. DOI: [10.1007/978-0-387-84858-7](https://doi.org/10.1007/978-0-387-84858-7).
- 473 [40] T. Schneider, “Methoden der automatisierten merkmalsextraktion und -selektion von sen-
474 sorsignalen,” Masterarbeit, Universität des Saarlandes, Saarbrücken, Deutschland, 2015.
- 475 [41] D. Kuhn, R. Müller, L. Hörauf, M. Karkowski, and M. Holländer, “Wandlungsfähige
476 Montagesysteme für die nachhaltige Produktion von morgen,” *wt Werkstattstechnik online*,
477 vol. 110, no. 09, pp. 579–584, Feb. 2020. DOI: [10.37544/1436-4980-2020-09](https://doi.org/10.37544/1436-4980-2020-09).