Artificial Intelligence supported Tool Chain in Manufacturing Engineering

ITEA 3 – 19027

Work package 4

Realistic and standardized production resource models

Deliverable 4.1

Concept & workflow of creation of models and automated selection of AI methods

Document type	: Deliverable
Document version	: 1
Document submission date	: 2023-04-04
Classification	: public
Contract Start Date	: 2021-01-01
Contract End Date	: 2024-02-29

Final approval	**Name**	**Partner**
Review Task Level	Martin Kratschmer	ISB
Review WP Level	Matthias Riedl	IFAK
Review Board Level	Kristofer Bengtsson	Volvo

Executive Summary

This document presents a comprehensive overview of an approach focused on the collection, preprocessing, and analysis of data from various use cases. The use cases covered in this document involve both time series data and image data, and the data collected from these sources is used to demonstrate a range of AI techniques. The methods shown are used to create models that can be used for functional simulation.

Initially, the document deals with methods for collecting and organizing data from these use cases, including the use of decision trees to support preprocessing. The decision tree is used to show which preprocessing methods can be used for the respective data.

The document also introduces the PAIS(R) model, which is a framework for understanding and comparing different AI methods. The model includes various phases such as identifying requirements, concept development, implementation and integration, validation and verification, and maintenance of AI systems.

In addition to the conceptual and design aspects of the approach, the document also includes a section on domain stories from various partners. These stories illustrate the applications of the data and AI techniques described in the document in real life and provide valuable insights into the potential impacts of these technologies as well as the roles and systems of the partners. In addition, the workflows and dependencies of partners and other work packages are described here.

Finally, the document concludes with initial design concepts to the user interface for the pipeline process from data acquisition, the use of data preprocessing, to the use of pipelines with AI methods. These approaches describe the required status and configuration elements for using the pipelines, to provide a clear and intuitive overview of the various steps in this process and to make it easier for users to access and use the data and AI methods described in the document.

Content

1. Introduction

In this document, the concept and design of the Workflows Get Real Measurement (GRM), Prepare Real Measurement, and the benefits of using AI methods (AIM) are investigated and defined. The target picture of WP4 is shown in Figure 1. The GRM will use the Use-Cases to capture real data. These data can reflect different behaviors of components, but they can also include energy measurements. This data should be preprocessed in a suitable form and then made available for AI methods. As a result, the generated AI models should be made available in suitable simulation formats such as FMU. They can then be validated in a suitable simulation environment and made available for practical usage. This document shows the provided data and how it can be collected. It also shows possible ways to preprocess data and then use AI methods to provide suitable simulation models for behavior. The concepts and designs described here should support the implementation of a platform in the next step, which will allow users to apply the different workflows for their purposes. The application scenarios are described in this document using domain stories to illustrate the use of the platform. These stories provide a clear picture of how the platform will be used in real-world situations and help to understand the benefits it can provide. Additionally, this document also covers initial UI approaches that will be implemented for the platform. These approaches are designed to provide a user-friendly and intuitive interface for users to interact with the platform.

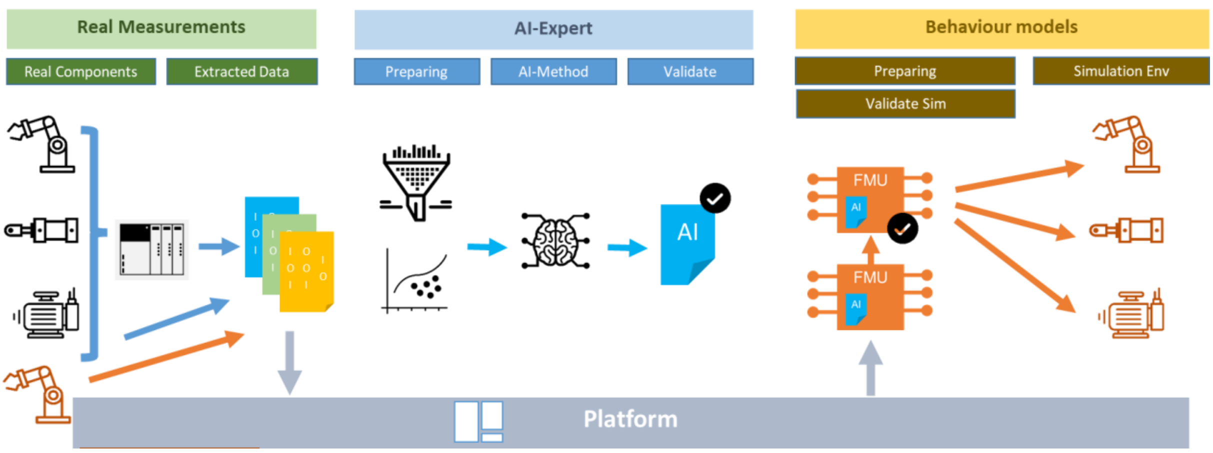

Figure 1 Target picture of WP4 in AIToC-Project

2. Get Real Measurements

2.1.1. Overview of Get Real Measurements

The GRM workflow deals with the description of data for existing use cases, the capture of data, and the use of the data marketplace, which will be implemented in work package 2.

In this chapter, the data for each individual use case is described. This includes information about the specific data points that are relevant to the use case, as well as any additional details or context that may be necessary for understanding the data.

2.1.2. Control and Process Signals (FFT Use Case)

This section explains the use case given by EKS InTec GmbH, which is a frequency converter controlling an industrial equipment, i.e. a turntable. The corresponding data is recorded and exported to the raw data format. A frequency converter is an electrical or electromechanical device that transforms one frequency of current to a different frequency of current. Before and after frequency conversion, the voltage is usually the same. The component is generally used to regulate the speed of motors. In the present use case, it is used to control the motor, which moves a turntable. The turntable is used to rotate a part on a platform so that a robot arm may perform tasks on the part. The frequency converter reads input signals from the PLC, processes the signals, and calculates the corresponding output signals. The PLC moves into an undesired state if output signals are not produced in a timely manner in response to input signals. The collected dataset contains an input file and an output file which includes control variables and process variables. In addition to the recorded control-and process data, a symbol table file is delivered to map a description to each input and output. The frequency converter reads the output signals from the PLC, processes the values and sends feedback of the current state via the output signals. Most of the data is qualitative in terms of Boolean values. The other part of the data is quantitative, which includes numerical values, representing a continuous behavior. The dataset has 3 cycles; each cycle takes 2 minutes. The data is collected every 50ms and for a period of 6 minutes. The data set has 7430 recordings or instances. The PLC IN data contains 15 variables. Out of the 15 variables 10 are control variables and are binary signals with Boolean values 1 is the timestamp and the remaining 4 variables are process variables with positive real numbers. The PLC OUT data contains 22 variables. Out of the 22 variables 17 are control variables and are binary signals with Boolean values, 1 is the timestamp and the remaining 4 variables are process variables with positive real numbers.

Timestamp	1.050TT1DU1TA01_F_OUT.PowerRemoved	1.050TT	1.050T	1.050T	1.050T	1.050T	1.050T	1.050T	1.050T	1.050TT	1.050TT1DU1	1.050TT1DU1TA01_OUT.ActualSpeed	1.050TT1DU1	1.050TT1DU1
2019-02-08T10:48:44.3110000Z	False	2	False	True	True	False	False	False	False	False	0	0	0	60
2019-02-08T10:48:44.3610000Z	False	2	False	True	True	False	False	False	False	False	0	0	0	90
2019-02-08T10:48:44.4110000Z	False	2	False	True	True	False	False	False	False	False	0	0	0	130
2019-02-08T10:48:44.4610000Z	False	2	False	True	True	False	False	False	False	False	0	0	0	150
2019-02-08T10:48:44.5110000Z	False	2	False	True	True	False	True	False	False	False	0	0	0	190
2019-02-08T10:48:44.5610000Z	False	2	False	True	True	False	True	False	False	False	0	3	65514	200
2019-02-08T10:48:44.6110000Z	False	10	False	True	True	False	True	False	False	False	0	65534	65491	210
2019-02-08T10:48:44.6610000Z	False	10	False	True	True	False	True	False	False	False	0	0	65514	220
2019-02-08T10:48:44.7110000Z	False	10	False	True	True	False	True	False	False	False	0	0	65520	220
2019-02-08T10:48:44.7610000Z	False	10	True	True	True	False	True	False	False	False	0	65533	70	220
2019-02-08T10:48:44.8110000Z	False	10	True	True	True	False	True	False	False	False	0	121	265	210
2019-02-08T10:48:44.8610000Z	False	10	True	True	True	False	True	False	False	False	3	257	236	190
2019-02-08T10:48:44.9110000Z	False	10	True	True	True	False	True	False	False	False	6	399	218	180
2019-02-08T10:48:44.9610000Z	False	10	True	True	True	False	True	False	False	False	8	530	226	170
2019-02-08T10:48:45.0110000Z	False	10	True	True	True	False	True	False	False	False	14	598	205	160
2019-02-08T10:48:45.0610000Z	False	10	True	True	True	False	True	False	False	False	21	730	242	160
2019-02-08T10:48:45.1110000Z	False	10	True	True	True	False	True	False	False	False	29	871	214	150
2019-02-08T10:48:45.1610000Z	False	10	True	True	True	False	True	False	False	False	39	1002	251	150
2019-02-08T10:48:45.2110000Z	False	10	True	True	True	False	True	False	False	False	44	1068	259	150
2019-02-08T10:48:45.2610000Z	False	10	True	True	True	False	True	False	False	False	62	1344	236	140
2019-02-08T10:48:45.3110000Z	False	10	True	True	True	False	True	False	False	False	69	1411	228	140
2019-02-08T10:48:45.3610000Z	False	10	True	True	True	False	True	False	False	False	83	1474	263	140
2019-02-08T10:48:45.4110000Z	False	10	True	True	True	False	True	False	False	False	107	1676	285	140
2019-02-08T10:48:45.4610000Z	False	10	True	True	True	False	True	False	False	False	116	1746	244	140
2019-02-08T10:48:45.5110000Z	False	10	True	True	True	False	True	False	False	False	135	1883	259	150
2019-02-08T10:48:45.5610000Z	False	10	True	True	True	False	True	False	False	False	155	1994	677	140
2019-02-08T10:48:45.6110000Z	False	10	True	True	True	False	True	False	False	False	175	2020	154	130
2019-02-08T10:48:45.6610000Z	False	10	True	True	True	False	True	False	False	False	196	1983	156	130
2019-02-08T10:48:45.7110000Z	False	10	True	True	True	False	True	False	False	False	226	2003	228	130
2019-02-08T10:48:45.7610000Z	False	10	True	True	True	False	True	False	False	False	226	2003	228	130

Figure 2 Sample of data from FFT Use Case

2.1.3. Material flow of a Production System (TWT Use Case)

The use case of TWT is the material flow of a lab-scale production system that handles and processes dummy workpieces. The main components of the use case are a production line, two robots with vacuum grippers, two separate conveyors, as well as an input and an output storage area. The setup of the material flow of a production system use case is depicted in the following figure.

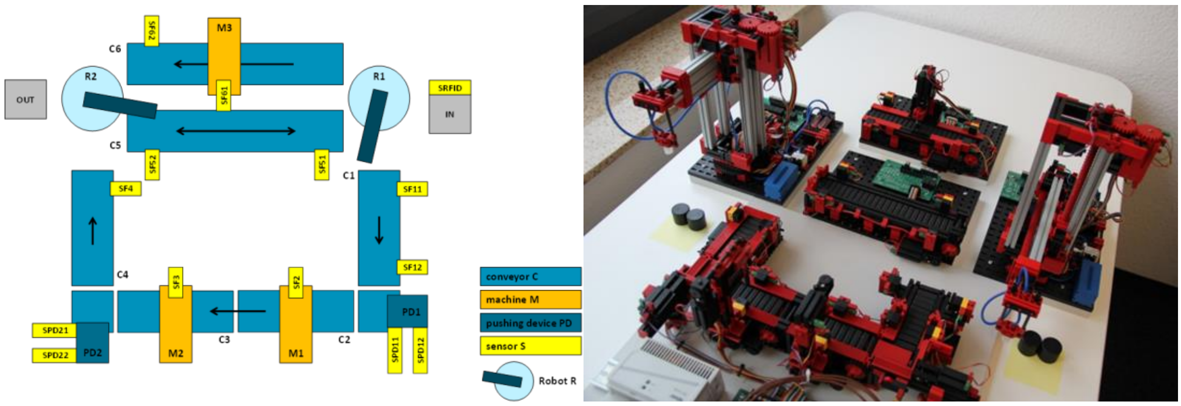

Figure 3 Functional sketch and photo of the material flow of a production system use case.

The production line has four conveyors arranged in a line which are connected by two pushing devices. Two machines are placed alongside the material flow to represent two operations performed on the workpieces. Furthermore, along the material flow are five Boolean light sensors to trace the workpieces through the production line. Additionally, four Boolean limit switches are available to detect the location of the pushing devices at their end positions. At one of the separate conveyors an additional machine is placed, that represents another operation to be performed on the workpieces. Both conveyors have each two binary light sensors for locating the workpieces at designated positions. Each robot has three degrees of freedom, i.e. vertical movement, horizontal movement and rotation. Each degree of freedom is moved by a separate motor with an encoder, as well as each degree of freedom has a limit switch to determine one of the two end positions. A vacuum gripper connected to a pump enables each robot to pick a workpiece and to transport it.

The collected datasets contain the input and output signals to three PLCs controlling the production system, i.e. the sensor and actuator signals. An extract from a collected dataset, specifically from the production line, is presented in following table. A collected dataset consists of a configuration file and a graphical depiction describing the used production system setup, a record key mapping file that maps an address port to each input and output signal, and the data in one or more csv-files.

1	timeStamp	slaveAddress	SF11	SF12	SF2	SF3	SF4	SPD11	SPD12	SPD21	SPD22	C1	C2	C3	C4	MS1	MS2
1095	2020-03-18T16:06:12.768199500+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	1	0	1	0	1	0	0	0	1	0	1
1096	2020-03-18T16:06:12.818206100+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	1	0	1	0	1	0	0	0	1	0	1
1097	2020-03-18T16:06:12.869197300+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	1	0	1	0	1	0	0	0	1	0	1
1098	2020-03-18T16:06:12.919207500+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	1	0	1	0	1	0	0	0	1	0	1
1099	2020-03-18T16:06:12.969361700+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1
1100	2020-03-18T16:06:13.019481400+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1
1101	2020-03-18T16:06:13.070359400+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	1
1102	2020-03-18T16:06:13.120496300+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0
1103	2020-03-18T16:06:13.171227400+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0
1104	2020-03-18T16:06:13.221354100+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	0	1	0	0
1105	2020-03-18T16:06:13.271501300+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
1106	2020-03-18T16:06:13.322225800+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
1107	2020-03-18T16:06:13.373214600+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
1108	2020-03-18T16:06:13.424256700+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	1	0	0
1109	2020-03-18T16:06:13.475223300+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0
1110	2020-03-18T16:06:13.526226400+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0
1111	2020-03-18T16:06:13.576249500+01:00[Europe/Berlin]	10.12.90.10	1	1	1	0	0	0	1	0	1	0	0	1	0	0	0
1112	2020-03-18T16:06:13.627209500+01:00[Europe/Berlin]	10.12.90.10	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0
1113	2020-03-18T16:06:13.678206+01:00[Europe/Berlin]	10.12.90.10	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0
1114	2020-03-18T16:06:13.728228500+01:00[Europe/Berlin]	10.12.90.10	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0
1115	2020-03-18T16:06:13.779209800+01:00[Europe/Berlin]	10.12.90.10	1	1	1	1	0	0	1	0	1	0	0	1	0	0	0

Table 1 Extract from a measured data set from the production line of the material flow of a production system use case (“1” is true and “0” is false).

In total, the production system contains 32 Boolean actuator signals (motor signals), 9 Boolean light sensors, 12 Boolean limit switch sensors, and 12 encoder sensors. A classification of different data signals (sensor and actuator signals) for the production line and a robot is given in following figure.

Sensors/Actuators	signal type	signal memory	max. measurement frequency (based on used input card)	quantity	description
production line					
sensors					
light sensor	digital	1 bit	3 ms/sample	5	is used to determine whether a product is in front of the sensor or not
reference switch	digital	1 bit	3 ms/sample	4	is used to determine whether the pushing device is at end position or not
actuators					
conveyor motor	digital	1 bit	-	4	mono-directional motor
machine motor	digital	1 bit	-	2	mono-directional motor
pushing device motor	digital	2 bits	-	2	bi-directional motor
robot					
sensors					
reference switch	digital	1 bit	3 ms/sample	3	is used to determine the end position of a rotational, horizontal or vertical movement
incremental encoder	analog	2 bytes	4 ms/sample	1	is used to determine the rotational movement
digital counter	digital	2 bits	0.2 ms/sample	2	can be used to determine a horizontal or vertical movement
actuators					
movement motor	digital	2 bits	-	3	bi-directional motor; for rotational, horizontal or vertical movement
compressor motor	digital	1 bit	-	1	part of vacuum gripper used to transport product
valve motor	digital	1 bit	-	1	part of vacuum gripper used to transport product

Figure 4 Classification of different data signals at the material flow of a production system UC.

2.1.4. Transport and Drilling a Workpiece Use Case

The demonstrator combines simple processes from manufacturing. A work piece is fictitiously drilled and then sorted according to its material. Figure 1 shows the involved components.

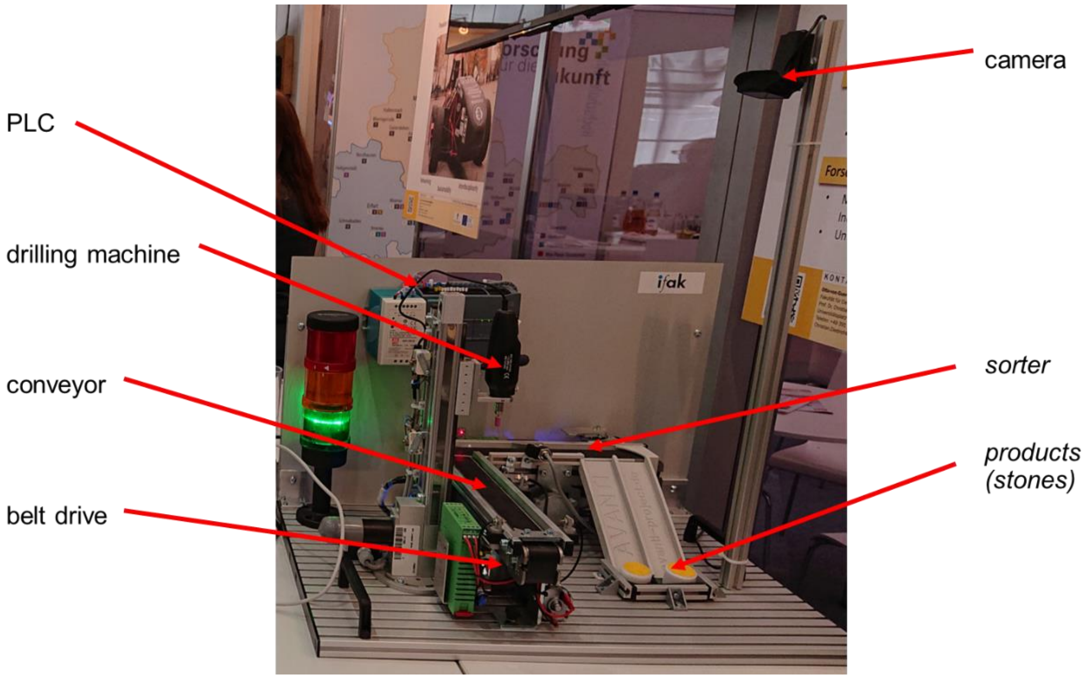

Figure 5 Components of the ifak demonstrator

The following processing steps are executed:

(01) Work piece comes into the production process
The work piece is manually inserted into the process. A photoelectric sensor detects the inserted work piece. The belt of conveyor 1 will be switched on.

(02) The work piece reaches the machine tool station.
A photoelectric sensor detects the work piece at the machine station. The belt of conveyor 1 will be switched off.

(03) The machine drills a hole
The machine is switched on, the drill is spinning. The machine is moving down to the position as defined for the deep of the hole in the work piece. After this, the machine is moving up. In the end, the machine stops spinning.

(04) The work piece is transported away from the machine.
If drilling is finished, the belt of conveyor 1 will be switched on.

(05) The work piece is move to a second conveyor.
At the end of conveyor 1 a distance sensor detects a coming work piece and switches on the belt at conveyor 2. The work piece moves from belt 1 to belt 2.

(06) The work piece is transported away from handover station.
If the distance sensor detects that the work piece is moved away, the belt at conveyor 1 is stopped.

(07) The work piece passes the material detector
the material detector checks the material of the work piece and classifies it to plastics or metal.

(08) Material sorter puts work piece on correct ramp
Depending of the material detection, the sorter moves the work piece to ramp 1 for plastics or ramp2 for metal. The belt of conveyor 2 is switched off after 3 second after sorting.

(09) The work piece can be removed manually.

The recorded dataset contains the inputs and outputs of the PLC. Except the time stamp in the first column, all data is of type Boolean. Figure 2 shows the structure of available data, formatted as CSV.

Timestamp	DI a.0	DI a.1	DI a.2	DI a.3	DI a.4	DI a.5	DI a.6	DI a.7	DI b.0	DI b.1	DI b.2	DI b.3	DI b.4	DI b.5	DI b.6	DI b.7	DQ a.0
2022-06-08 15:17:45.906592	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False
2022-06-08 15:17:56.367304	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	True
2022-06-08 15:17:56.374783	False	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	True
2022-06-08 15:17:56.675231	True	False	False	True	False	False	False	False	False	False	False	False	False	False	False	False	True
2022-06-08 15:17:57.797220	True	True	False	True	False	False	False	False	False	False	False	False	False	False	False	False	False

Figure 6 Format of the data file

The file consists of 2 octets of inputs and 2 octets of output data. In fact only only 14 bits of inputs and 10 bits of output are used, the remaining bits DIb.6, DIb.7 and DQb.2 - DQb.7 are unused and redundant.

2.1.5. Energy consumption behaviour of spot welding machine

Although TOFAŞ does not provide Official Use Case, ARD brings an additional use case to contribute project works with the help of TOFAŞ. Our use case focuses on developing an energy consumption behavior of spot welding machine. The relevant data is provided by TOFAŞ that are taken from real factory plant. The following figure depicts the process, where robots are being utilized to accomplish spot welding operations in their real plant.

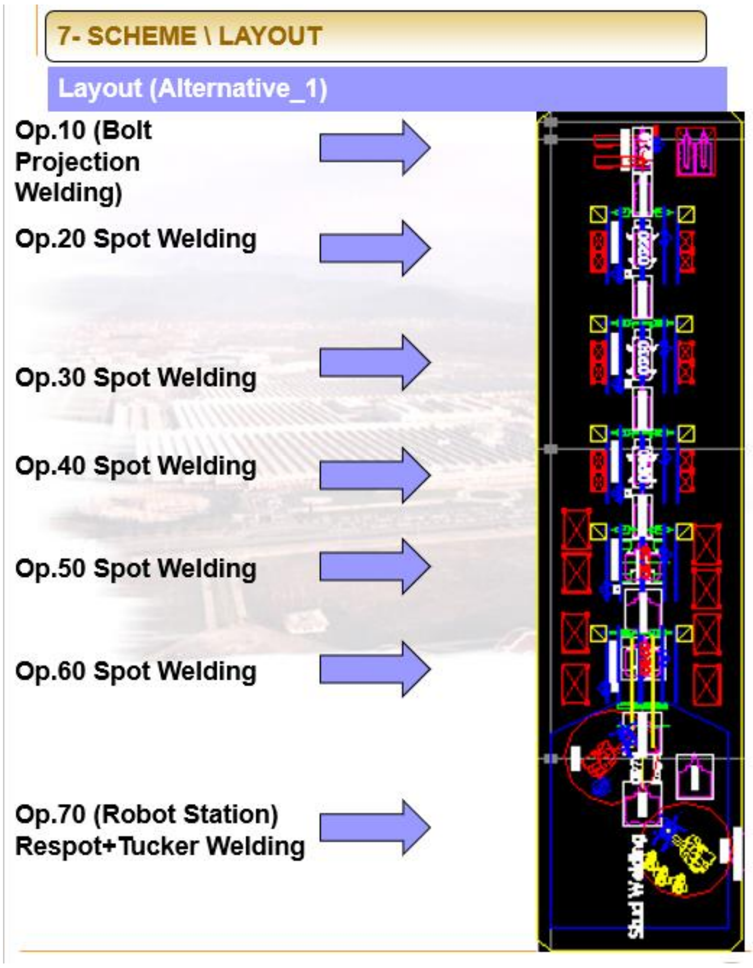

Figure 7: Layout of spot welding machines

Every single operation has its own id, which is linked to information about the place/positions and other relevant info. Spot welding device requires some inputs in order to achieve its expected results, which contains the type of sheets, thickness, repeat, force, etc. in addition to other set parameters.

Set parameters contains the current and duration for pre-weld operation, current and time for welding operation, expected energy consumption in joule.

The output of this operation is collected as the measured data. Measured data contains the same parameters as of “set parameters” but with actual results measured (real data) during the operation.

Data parameters to be used are;

- Sheet thickness,
- Electrode diameter,

- Number of times the electrode used
- Force
- Current
- Welding times
- Energy consumption

Here is an example real data obtained from TOFAŞ. Data is contained in three different “csv” file. These are

(i) Actual Parameters, whose sample data is as follows;

```
#Read the CSV File into df
dfAp = pd.read_csv('TOFAS_welding_parameters_analysis-Actual Parameters.csv')
```

```
dfAp.head()
```

	TimerName	Kaynak ProgNo	Spot Name	Punta Sayisi	Uç Kullanım Yüzdesi	voltage Actual Value	voltage Ref Value	current Actual Value	current Reference Value	weld Time Actual Value	weld Time Ref Value	energy Actual Value	energy Ref Value	resistance Actual Value	resistance Ref Value
0	BSR010WB01	2	6790_0_00	45	76.62	1.06	1.03	8.24	8.39	400	400	3549.324	3544.768	133	127
1	BSR010WB01	1	6788_0_00	46	76.63	1.09	1.09	8.48	8.47	407	400	3814.062	3732.033	134	134
2	BSR010WB01	5	6812_0_00	47	76.65	1.05	1.04	8.27	8.39	409	400	3596.276	3560.329	129	126
3	BSR010WB01	6	6818_0_00	48	76.67	1.03	1.02	8.41	8.39	420	400	3678.003	3501.718	127	125
4	BSR010WB01	7	6824_0_00	49	76.69	1.05	1.03	8.30	8.38	420	400	3693.769	3545.885	131	126

Figure 8 Actual data of data from TOFAŞ

(ii) Product Info file containing data about Line, Station, Robot beeing used, Electrode Type, welding program id, material codes, thicknesses. Sample data is as follows;

```
dfPI= pd.read_csv('TOFAS_welding_parameters_analysis-Product Info.csv')
```

```
dfPI.head()
```

	Hat	İstasyon	Robot No	Ekektrod tipi (Ø - mm)	Spot Name	Kaynak Program No	Malzeme1	Malzeme2	Kalınlık1	Kalınlık2
0	DX FIANCATE	10	R1	6	6812.0	5	FEP06	FE P04	0.65	0.7
1	DX FIANCATE	10	R1	6	6818.0	6	FEP06	FE P04	0.65	0.7
2	DX FIANCATE	10	R1	6	6824.0	7	FEP06	FE P04	0.65	0.7
3	DX FIANCATE	10	R1	6	6830.0	8	FEP06	FE P04	0.65	0.7
4	DX FIANCATE	10	R1	6	6838.0	9	FEP06	FE P04	0.65	0.7

Figure 9 Sample of product information of data from TOFAŞ

(iii) Set Parameters containing Program No, Welding Repitation, and Force in kN. Sample data is as follows;

```
dfSp= pd.read_csv('TOFAS_welding_parameters_analysis-Set Parameters.csv')
```

```
dfSp.head()
```

	TimerName	Kaynak ProgNo	Kaynak Tekrar Sayısı	Kuvvet (kN)
0	BSR010WB01	1	1	2.0
1	BSR010WB01	2	1	2.0
2	BSR010WB01	3	1	2.4
3	BSR010WB01	4	1	2.4
4	BSR010WB01	5	1	2.0

Figure 10 Sample of set parameters of data from TOFAŞ

Our aim is to predict energy consumption of spot welding operation using an AI Model, trained with the real data we collected. For this purpose, we first need to preprocess and prepare data for training in AI model. To do this, we first joined/merged the tables, and removed unnecessary columns. Then, split data into training (%80) and test (%20) dataset, and used multi-variable LinearRegression method with/without Polynomial Features. The results found very accurate, for both methods. Although the Polynomial method slightly improves accuracy and reduces errors, there does not seem to be a need to use the Polynomial Feature, considering this small change in improvement vs. increased computation time.

```
score = r2_score(y_test, y_fit)
print("The accuracy of our model is {}%".format(round(score, 10) *100))
score = mean_absolute_error(y_test, y_fit)
print("The Mean Absolute Error of our Model is {}".format(round(score, 2)))
score = mean_squared_error(y_test, y_fit)
print("The Mean Square Error of our Model is {}".format(round(score, 2)))
```

```
The accuracy of our model is 99.96203836999999%
The Mean Absolute Error of our Model is 8.58
The Mean Square Error of our Model is 118.02
```

```
score = r2_score(y_test, y_pred)
print("The accuracy of our model is {}%".format(round(score, 10) *100))
score = mean_absolute_error(y_test, y_pred)
print("The Mean Absolute Error of our Model is {}".format(round(score, 2)))
score = mean_squared_error(y_test, y_pred)
print("The Mean Square Error of our Model is {}".format(round(score, 2)))
```

```
The accuracy of our model is 99.92012758999999%
The Mean Absolute Error of our Model is 12.19
The Mean Square Error of our Model is 248.32
```

Figure 11 First results of preprocessed data

Our next job is to train different models with different features, and compare them to select the best AIM. And finally, we are to export these models as FMU.

2.1.6. Product quality monitoring

Our demonstrator represents data analysis for online product quality monitoring based on camera system. The input data is a set of images representing an elastic circuit board at various stages of manufacturing – after printing of the conductive ink, and after component installation on that foil. The first stage aims at controlling quality of the conductive layer, identifying faults, missing connections, or short circuits. This data will be used for automatic recalibration of the printing process. The second stage is analysis of component placement - correction of the component orientation on the circuit, and positioning accuracy, including test if all the component legs make connection to the circuit on the film and if there are no short circuits or excessive adhesive, that is conductive and could cause potential faults in the system.

The data is streamed from the camera to the Data Marketplace (developed as a result of WP2), then data is fetched by the Data Marketplace connector to the AIM platform, where processing pipeline is designed utilizing individual image processing modules. The final result is again pushed to the Data Marketplace where all pipeline parameters are saved together with the relations to the input and output data.

In addition to image processing in quality monitoring, we consider the use case where online production data is created as time series data, collected by the Data Marketplace and analyzed using AIM platform to update digital twins of individual production machines. The data from the digital twins of the machines will be then utilized in the WP5 – layout planning of the factory and estimation of the production throughput and all associated costs.

2.1.7. Portal Robot Use Case

Our use case demonstrates the use of a 3-axis portal robot whose primary application is automated kitting operations for the Volvo Truck Final Assembly Plant. Other potential applications include clinching and punching operations. The Portal Robot has a working volume of 800mm x 800mm x 500mm. Each axis is driven by an absolute encoder servo motor and controlled by a PLC. The PLC sends a signal to move to a particular position and continuously gets feedback on the current position of the servos. While the PLC can define parameters like maximum velocity and acceleration, the servo controller controls the motor.

The position to which the PLC wants to move the servo to is initially written as a double. A Boolean enables variables of the servo when it is required to move. When the servo starts moving, the PLC requests feedback from the servo regarding its position every 4ms scan cycle. When the servo reaches its final position, it writes a Boolean to the PLC, indicating it has reached its destination.

The variable state for an operation is depicted in Figure 12. The time is the first column and is a reference for all the other columns. The three axes' actual positions are shown in the next three columns, followed by their target positions with the data type as double. They are followed by six Boolean variables, the first three being the enable signal for the three-axis and the completion operation signal for the three axes.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Timestamp(ms)	mwXaxisPos	mwYaxisPos	mwZaxisPos	mwXPos	mwYPos	mwZPos	mxExeXaxis	mxExeYaxis	mxExeZaxis	mxDoneXaxis	mxDoneYaxis	mxDoneZaxis
2	21474836484	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
3	21474836488	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
4	21474836492	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
5	21474836496	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
6	21474836500	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
7	21474836504	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
8	21474836508	161.2470093	100.0250015	610.0010376	100	100	610	0	0	0	0	0	0
9	21474836512	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
10	21474836516	161.2480011	100.0240021	610	100	100	610	0	0	0	0	0	0
11	21474836520	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
12	21474836524	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
13	21474836528	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
14	21474836532	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
15	21474836536	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
16	21474836540	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
17	21474836544	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
18	21474836548	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
19	21474836552	161.2480011	100.0250015	610	100	100	610	0	0	0	0	0	0
20	21474836556	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
21	21474836560	161.2470093	100.0250015	610.0010376	100	100	610	0	0	0	0	0	0
22	21474836564	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
23	21474836568	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
24	21474836572	161.2480011	100.0250015	610	100	100	610	0	0	0	0	0	0
25	21474836576	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
26	21474836580	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
27	21474836584	161.2470093	100.0250015	610	100	100	610	0	0	0	0	0	0
28	21474836588	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0
29	21474836592	161.2480011	100.0250015	610	100	100	610	0	0	0	0	0	0
30	21474836596	161.2470093	100.0240021	610	100	100	610	0	0	0	0	0	0

Figure 12 Format of data file from portal robot.

2.2. Interrelation to AIToC-Data-Marketplace

Maintaining data integrity and managing datasets is realized through a centralized Data Marketplace. Every client of the framework can connect to Data Marketplace to fetch or push data to it. While interacting with the data marketplace, data formats are converted as required by the server based on internal storage format and format requested by the client. This approach allows also for interoperability with various tools limiting need for data mangling outside the data management platform. In addition, as a result of centralization, concurrent access to same datasets is possible as well as concurrent data modification with partly automatic conflict resolution. Access to the framework requires privilege verification and user authorization. This allows to create fine-grained access layer for the data, that should be shared between multiple entities. The Active Registry access layer of the Data Marketplace provides methods for verification user identity and therefore access rights, then allowing user to query for available data and if access rights permit it, also creating new datasets. Example communication is presented in Figure 13.

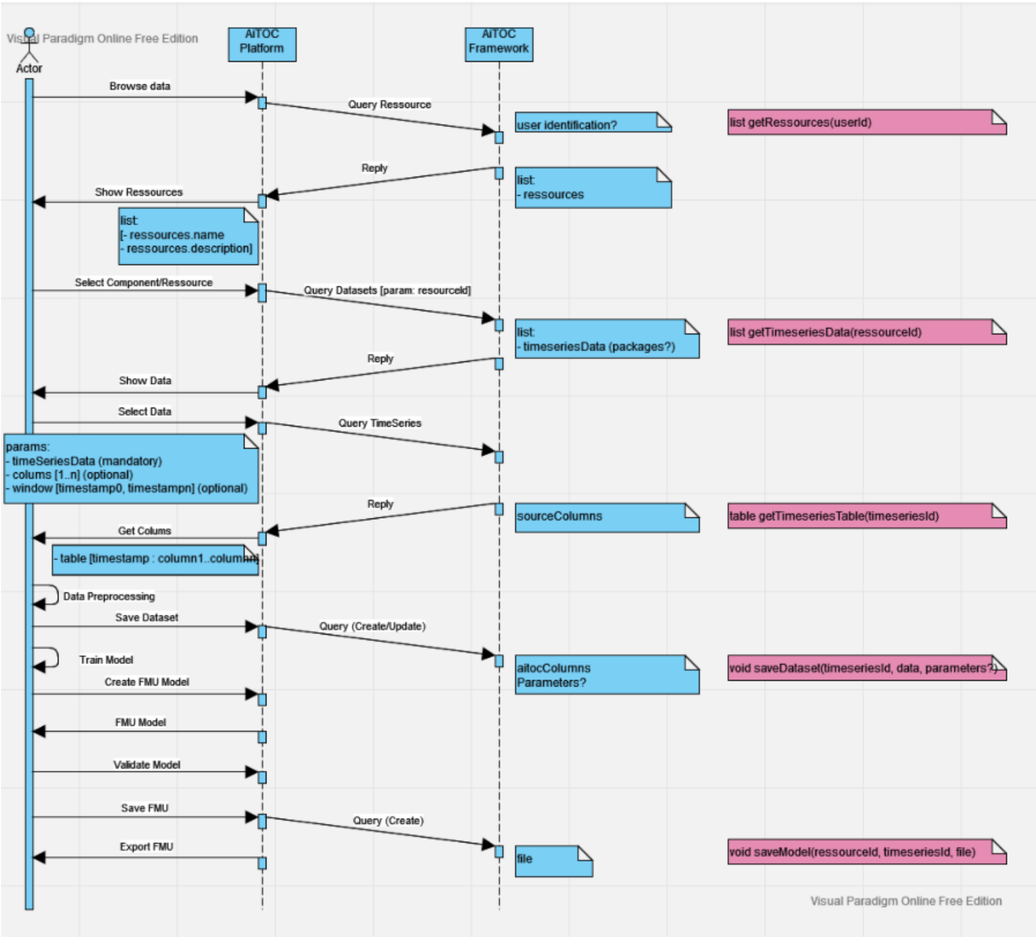

Figure 13 Sequence diagram for interrelation to data-marketplace

3. Prepare Real Measurements

3.1. Overview PRM

Preprocessing is a crucial step in many AI and machine learning methods because it helps to improve the quality of the input data. This can help to increase the accuracy of the model and make it more efficient. Preprocessing can include a variety of tasks, such as cleaning and formatting the data, scaling and normalizing the values, and transforming the data in other ways to make it more suitable for the specific AI or machine learning method that will be used. By preprocessing the data, we can help to ensure that the AI or machine learning model has the best possible chance of learning and making accurate predictions.

3.2. Generic Decision Tree (collected for time-series data)

3.2.1. Overview Generic Decisison Tree

In this chapter, a decision tree will be used to represent the sequence of data preprocessing methods. Within the data preprocessing, we start by checking the incoming data and at the end of the tree, we obtain a set of prepared data that can then be used for AI applications. In Figure 14, we see the decision tree in a summarized representation. This shows the categorization of the preprocessing methods and at which points the decision is made on whether to use this group of methods.

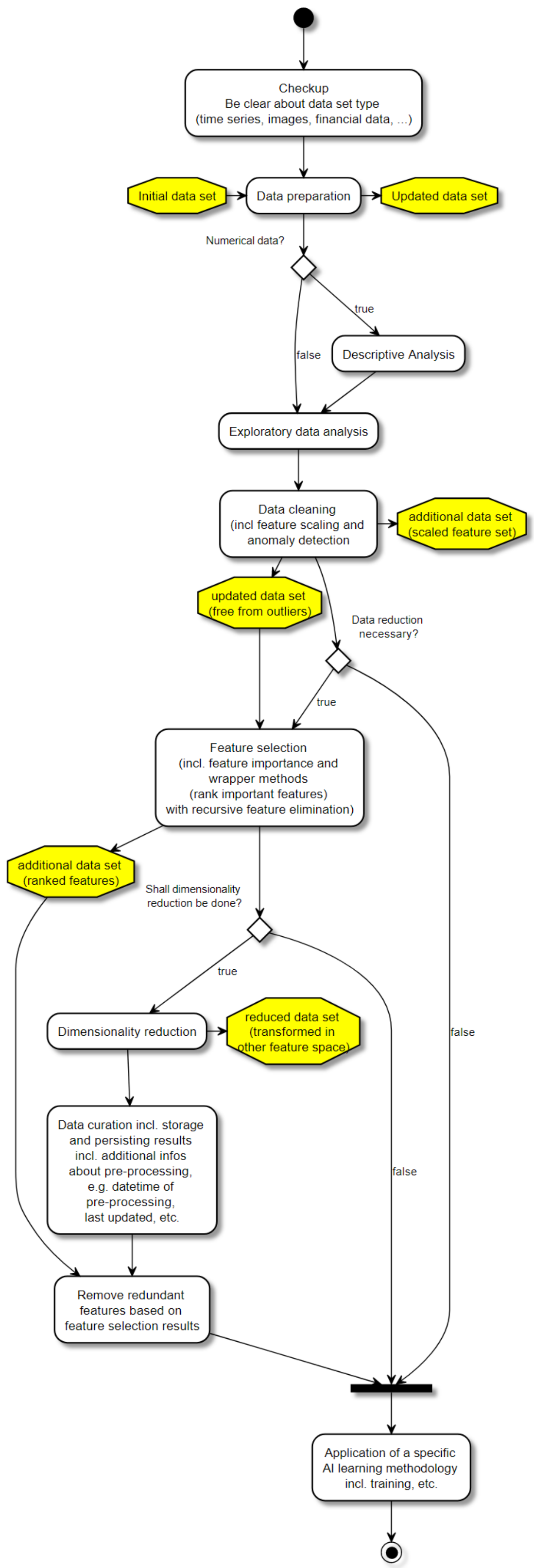

Figure 14 Overall view of the decision tree for selecting the appropriate preprocessing methods

3.2.2. Exploratory data analysis

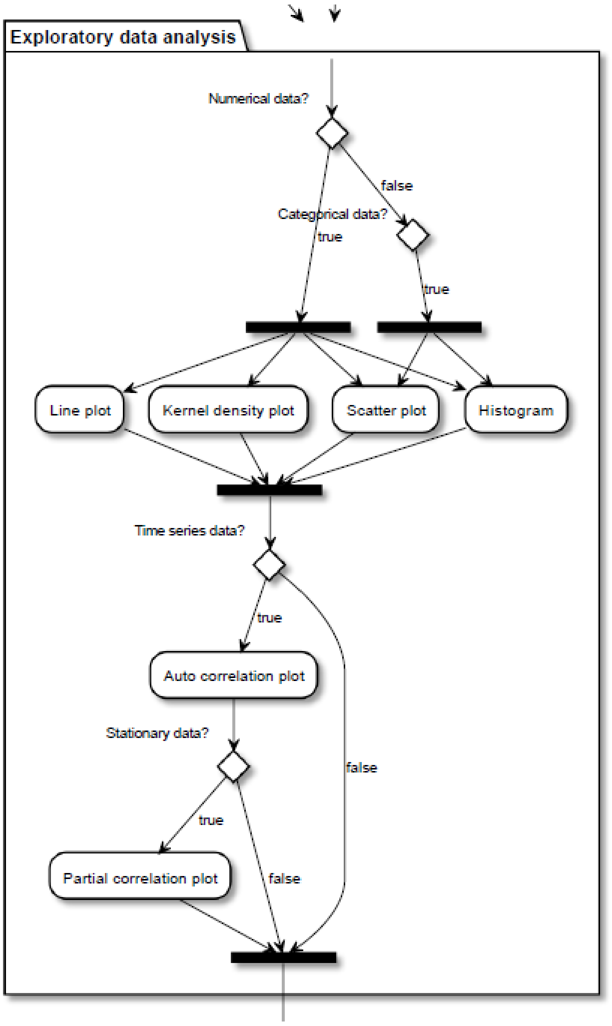

Figure 15 Detailed view of exploritory data analysis

Exploratory Data Analysis (EDA) is a way of evaluating datasets in order to summarize their key properties, which commonly involves the use of statistical graphics and other data visualization techniques. It makes it easier to understand and extract useful insights. In the present use case, the data has many features, and a visual representation will help in understanding the temporal structure. All the techniques in the EDA have been implemented or achieved through matplotlib and seaborn libraries.

The main advantages of performing an EDA on the raw data are:

- Getting a better understanding of the raw data
- Identifying various data patterns
- Getting a better understanding of the problem statement

In this work, histogram, kernel density plot, line plot and correlation plot have been incorporated to study the temporal structure of the data.

3.2.3. Data cleaning

Data cleaning (Figure 16) is the process of identifying and correcting or removing inaccurate, incomplete, or inconsistent data in a dataset before it is used for analysis or deployment in a machine learning application. This is particularly important when working with time series data, as faulty or inconsistent data can lead to inaccurate or flawed analysis or predictions. Some steps that can be taken in cleaning time series data include removing duplicates, filling in missing values, correcting inconsistent data, and removing outliers.

Feature scaling is a preprocessing technique that standardizes the independent features in a dataset by handling and scaling data with significantly changing values. This ensures that each feature is equally important and easier for machine learning algorithms to analyze. In this work, normalization and standardization are used for feature scaling.

Anomaly detection is the process of identifying unusual items or events in a dataset that differ from the norm. Anomalies can occur rarely and have significantly different features from normal instances or be outliers that are distant from other observations. Smoothing based anomaly detection algorithms and constraint-based outlier detection are used in this work for anomaly detection. Outliers can increase error variance and reduce the power of statistical tests and can bias and influence results.

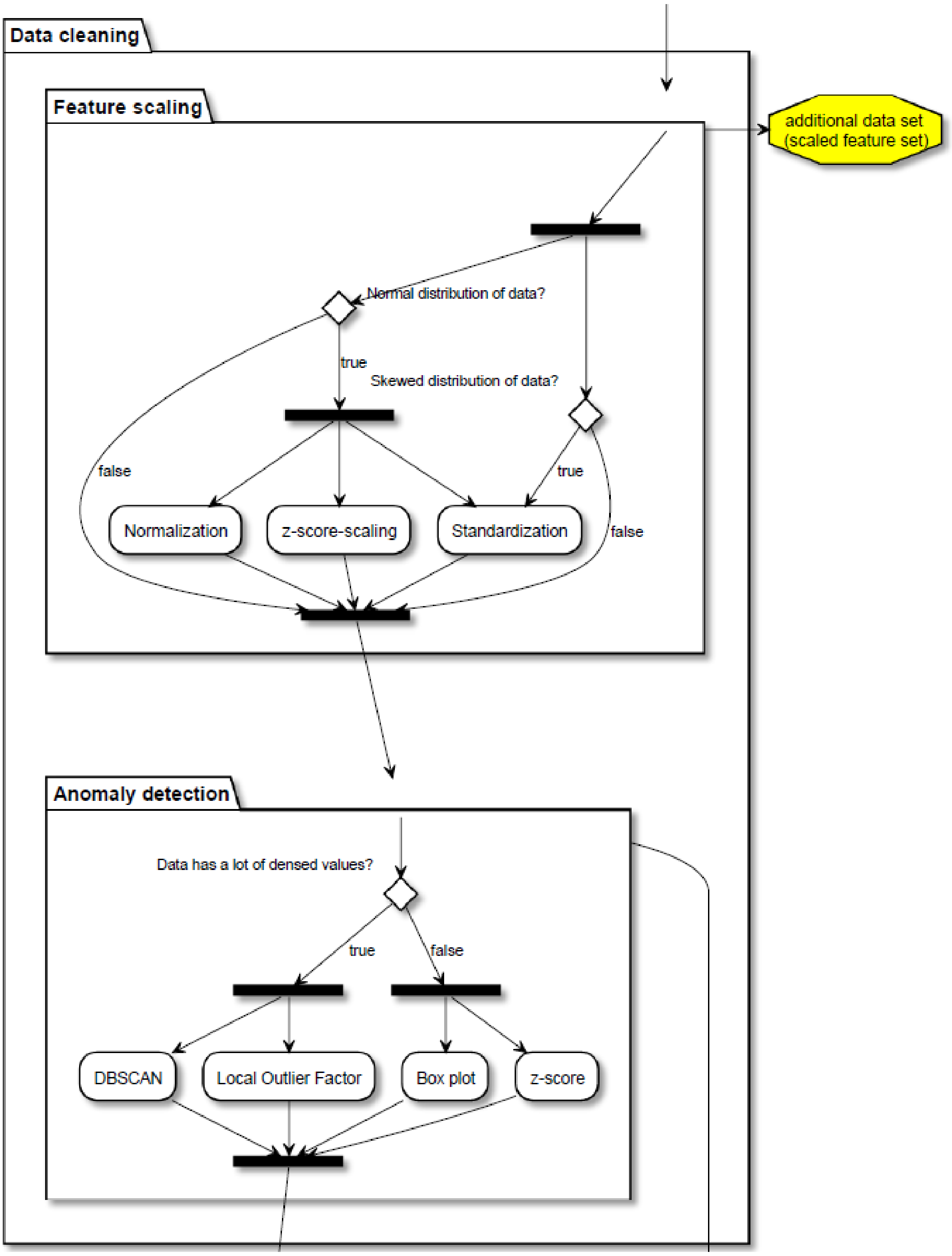

Figure 16 Detailed view of data cleaning

3.2.4. Feature selection

Feature selection is a method used in data reduction to identify and remove redundant or less important features from a dataset. This process can be done automatically or manually and has the goal of simplifying models, improving data mining speed, and producing clean, understandable data. The main advantages of feature selection include reduced training time and memory usage, reduced overfitting, improved accuracy and reliability, and easier interpretation. However, it should be noted that these techniques may sometimes lead to undesired results if not considered properly, as they can remove relevant features.

Feature selection techniques can be divided into two main categories: filter methods based on statistical tests and wrapper methods based on feature importance. Wrapper techniques evaluate the importance of features in a dataset based on the prediction performance of a predetermined machine learning algorithm. The Recursive Feature Elimination (RFE) method is a wrapper technique that iteratively removes the least important features based on the learning performance of a given machine learning algorithm. This method is applied in a use case where the dataset is divided into control and process variables, and the machine learning algorithm is run on the control variables against the process variables to learn the pattern of the dataset. Wrapper methods generally result in better predictive accuracy than filter methods.

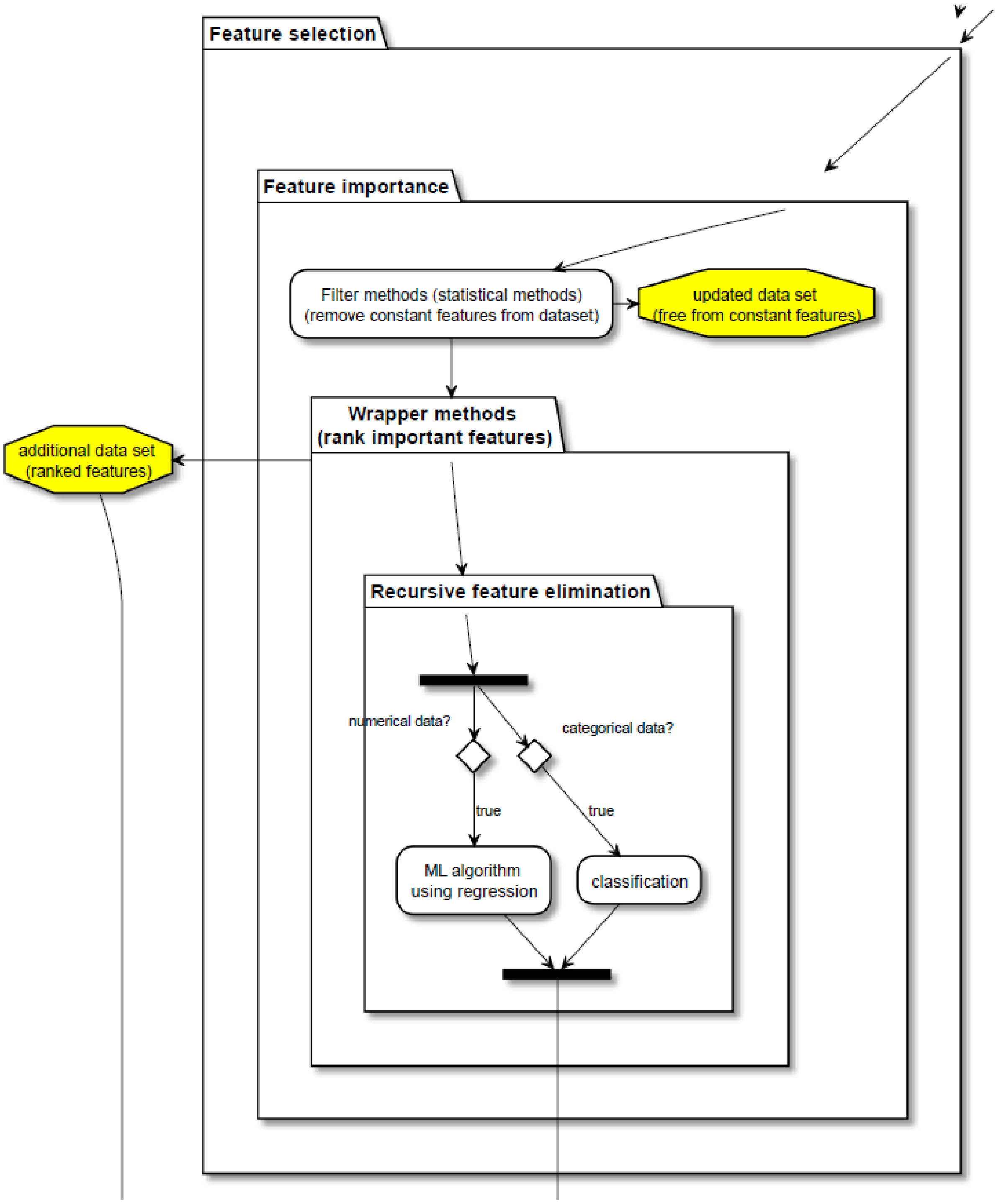

Figure 17 Detailed view of feature selection

3.2.5. Dimensionality reduction (optional)

Dimensionality reduction is the process of reducing the number of variables in a dataset while preserving as much variation as possible in the original dataset. It is a data preprocessing technique that reduces the number of dimensions in a dataset. The main advantages of dimensionality reduction include reduced time and storage space requirements, improved interpretation of machine learning model parameters, and easier visualization of data. However, it can also lead to some data loss. There are two types of dimensionality reduction techniques: linear and non-linear. In this work, only linear methods were applied because the data was linear and sequential, and only Principal Component Analysis (PCA) was used because it was found to be the most suitable for the use case.

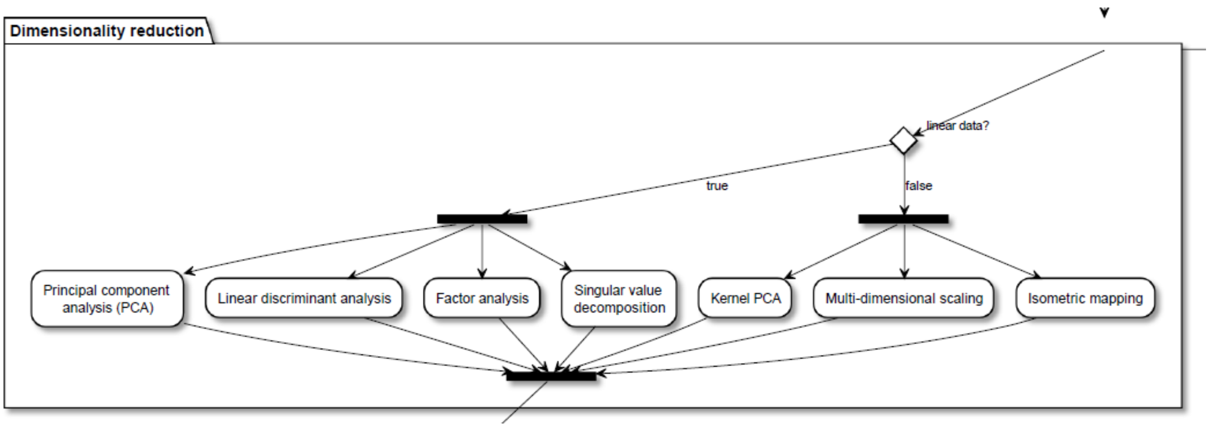

Figure 18 Detailed view of dimensionality reduction

3.3. Prepocessing methods (collected for image data)

Image processing methods require specific base data preparation. In many cases the following procedures are used:

- Image filtering – targeted at removing noise
- Color transformations – changing RGB image to HSL or HSV channels
- Thresholding – creating binary image mask based on single or multiple thresholds
- Image segmentation
- Morphological transformations
- Feature extraction
- Object detection
- Object labelling
- Image to image transition

Image processing methods aim at either conversion of an input image to another image or set of images, that are easier to process, or extraction of data from an image, for example locations of specific objects or detection of objects orientation, scale or other properties, like color, texture, etc.

4. AI-Methods

4.1. PAISE® model

4.1.1. Overview

The second edition of the DIN/DKE standardization roadmap Artificial Intelligence [7] calls for a systematic approach to the use of AI methods based on a process model. One representative of this is PAISE® [6] depicted in Figure 19. To develop and specify the AI pipeline and platform of AIToC in a systematic way, the PAISE® model was used.

“The process model for AI systems engineering, PAISE® for short, was developed at the Competence Center for AI Systems Engineering (CC-KING). PAISE® comprises the systematic and standardized development and operation of AI-based system solutions. Approaches from computer science and data-driven modeling are combined with those from traditional engineering disciplines, such as systems engineering.” [6]

The PAISE® process model is specifically designed with AI-System Engineering in mind.

It allows for modularization and parallel development.

The PAISE® model breaks down into several process steps. The realization of these steps in the context of the AI platform of AIToC will be explained in the subsequent chapters.

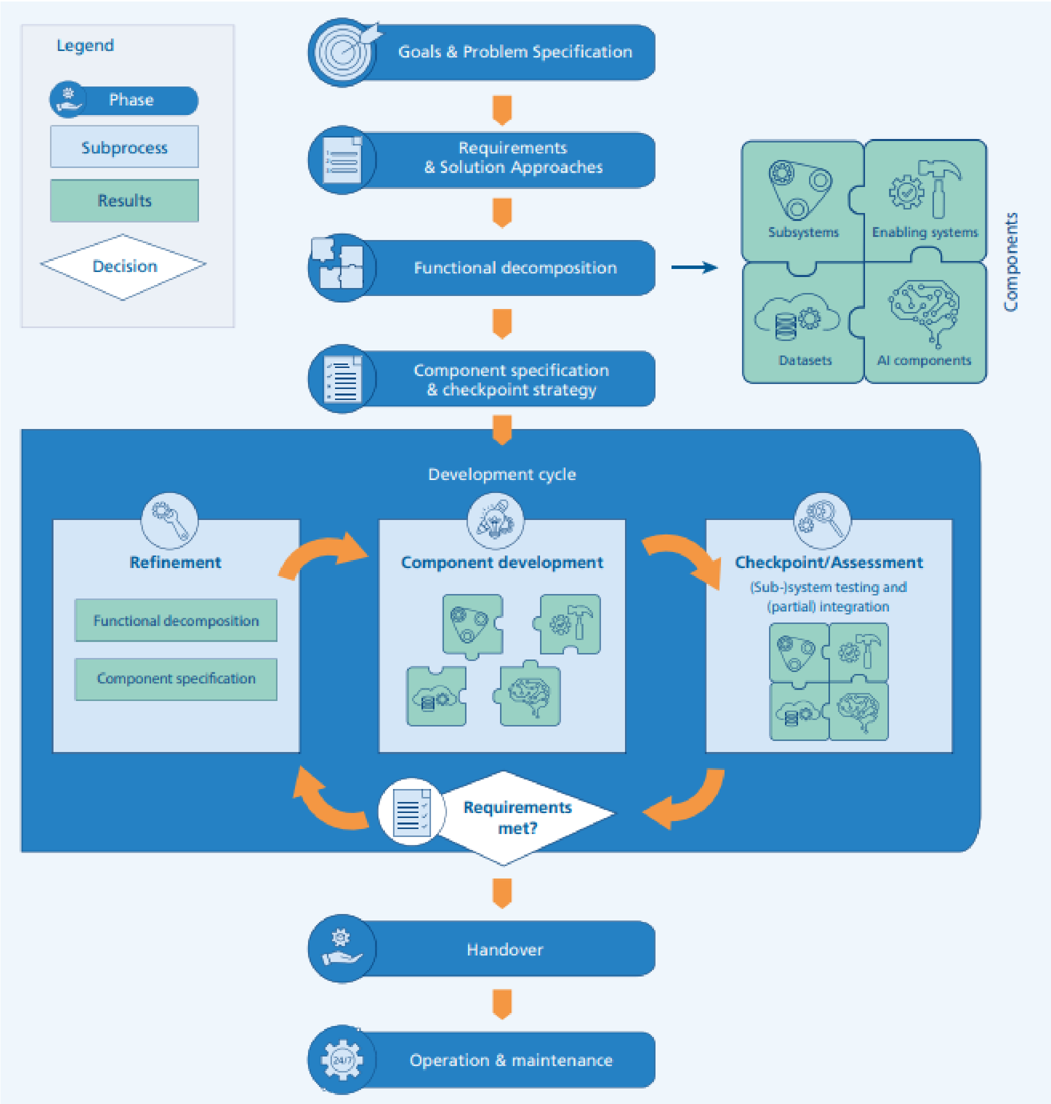

Figure 19 Overview of the different parts of the PAISE® model (Image source: [PAISE])

4.1.2. Step 1: Goals and Problem Specification

In the first step, the goals to be achieved with our AIToC AI platform are defined.

The usage of the AIToC platform and its possibility to hide the complexity of AI resp. the application domain from the domain expert, data scientist and AI expert is one of the key goals of the AIToC project.

The goals are driven by the use cases which are described by means of their relevant data in chapter **Error! Reference source not found.**. They range from the learning and packaging of complex behavior models of production components to the prediction of energy consumption data. The realization of the different use cases and the corresponding user interactions are described in chapter **Error! Reference source not found.**.

A common characteristic of the use cases is the modeling of a deterministic but complex system without knowledge about its internals. Most of the use cases have in common that they deal with time series data. Time series prediction is the task of predicting future values of a time-dependent variable based on its past values. Time series data is found in many fields, including finance, economics, and industrial manufacturing, and it is often characterized by patterns and trends that evolve over time or are cyclical in nature. These prediction models can then be used to implement fully functional (behavioral/energy) models (digital twins) of the respective machines or automated processes for simulation. The semi-automatic generation of such models by modern machine learning approaches allows for a much more efficient model development process compared to classical approaches, based on e.g. physics simulation or manually implemented algorithms.

4.1.3. Step 2: Requirements and Solution Aproaches

In the second step the requirements for the overall system, our AIToC AI platform, are analyzed and possible steps towards a solution for implementation are defined.

One main requirement is to hide especially the complexity of AI resp. the application domain from the domain expert, data scientist..

The different roles who are involved in the AIToC platform and its usage have different requirements which have to be taken into account for the designed solution to meet their needs. The data provider (normally the customer) is responsible for supplying the raw data. This might be e.g. a plant operator. The domain expert is the responsible domain expert who is capable of interpreting and exploring the collected data from the data provider and possesses a deep domain know-how. The Data Analyst is responsible for the building and parametrization of a processing pipeline which consists of different AIToC AI platform modules to be included in this pipeline. The AI-Expert is responsible for the design and implementation of the Machine Learning Models and their integration in the AIToC AI platform architecture.

Concerning feasible AI methods, the use cases and available data presented in chapter **Error! Reference source not found.** are focusing on time series prediction.

Modern deep learning methods have proven to be effective for time series prediction. Deep learning models, such as long short-term memory (LSTM) networks and convolutional neural networks (CNNs), can learn complex patterns and dependencies in the data and make accurate predictions. However, there are several challenges that must be addressed when using deep learning for time series prediction.

One challenge is the need to properly preprocess and prepare the data for input into the deep learning model. This may involve smoothing, scaling, and resampling the data to make it more suitable for training and prediction.

Another challenge is the selection of the appropriate deep learning model architecture and hyper parameters. There are many different deep learning models and configurations that can be used for time series prediction and finding the optimal combination can be a time consuming and iterative process (see chapter **Error! Reference source not found.**).

Finally, it is important to carefully evaluate the performance of the prediction model to ensure that it is making accurate predictions.

4.1.4. Step 3: Functional Decomposition

During the third step, the functions of the overall system are initially broken down into subsystems. In the present case, the AIToC AI platform was divided into the following subsystems (see Figure 20):

- Data Import
- Data Preprocessing (preparation of the raw data for the AI-training pipeline, e.g. outlier/anomaly detection, value normalization, type conversions, …)
- AI Model Training
- AI Model Export (export of the trained models, e.g. translation of the generated AI Model into a functional mockup unit/FMU)

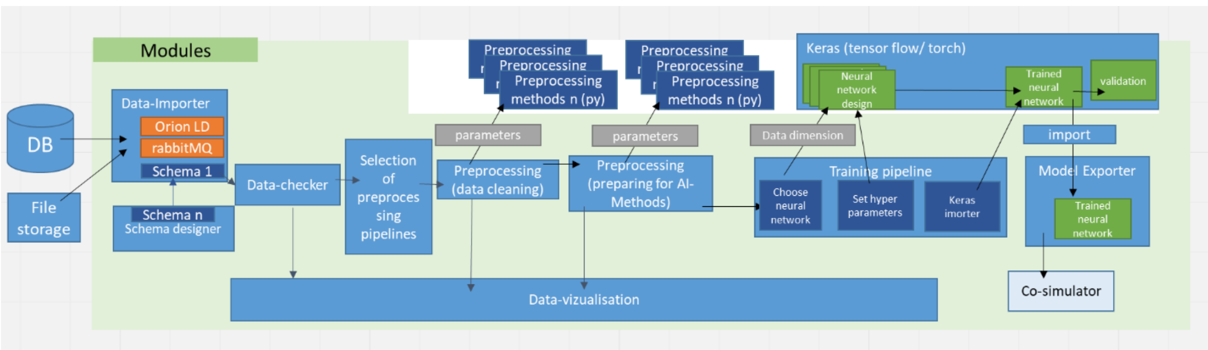

Figure 20 Overview of the subsystems within the AIToC AI platform

4.1.5. Step 4: Component specification & checkpoint strategy

In step 4, the preliminary version of the component specification is created. These solution approaches are reviewed and refined during the development cycle phase. The list of intended and mandatory modules of the AIToC AI platform (including their description and supporting underlying technologies) is depicted in Table 2 List of AIToC AI platform modules

Parent module	Module	Description	Technologies
	Data importer	Application for import data from one format or system into another	
Data importer	**Data structure parser**	Interpretation of e.g. AutomationML to determine the "path of the data"/"metadata"	AutomationML, Json, ...
Data importer	File importer	Import data from a file	CSV, HDF5, ...
Data importer	AIToC-Data-Marketplace Conector	Import and Export data on Data-Marketplace	tbd
Data importer	**DB connector**	Import and export data to certain data bases	InfluxDB, MariaDB (SQL), timescaleDB, postgres (noSQL
Data importer	**Message broker connector**	Publish and subscribe on certain message broker	rabbitMQ, OrionLD, Kafka...
Data importer	**Schema designer**	Configuration of the data structure	RDF, OWL
Data importer	**Data checker**	Proof imported data on the base of used schema	tbd
Data importer	**Data backbone**	Import data, publish and subscribe data	Kafka
	Data visualization	Data can be visualized in diagrams or tables	matplot, graphana, seaborn
	User Interface	Helps the user to select, configure, and perform settings on the platform for their pipelines.	Dear ImGui, NodeRed ...
User Interface	Pipeline selector	Filter and select prepared pipeline and suggestion on base of the imported data	tbd
	Pipeline templates	Reusable pipelines	gRPC, Docker, Kubernetes, AR
pipeline templates	Preprocessor (Data-integration)	Reusable preprocessing pipelines	NumPy,Pandas
Preprocessor (Data-Integration)	Preprocessing methods (Data -cleaning/ -reduction/ -transformation)	Individual preprocessing methods	tbd
pipeline templates	Preprocessor (AI)	Reusable preprocessing for AI pipelines	tbd
Preprocessor (AI)	Preprocessing methods (AI specific)	Individual preprocessing for AI methods	tbd
pipeline templates	Training pipeline	Reusable AI pipelines	tbd
Training pipeline	AIM-Tool connector	Access to e.g. NN-Tool and transfer preprocessed data and import trained model to platform	tbd
Training pipeline	AIM-Designer	Configure, select and parametrize e.g. the neurenal networks	KERAS (Tensor Flow, Torch)
Training pipeline	AIM-Trainer	Execution of the training and validation of the trained model	KERAS (Tensor Flow, Torch)
	Model exporter	Export it in common simulation format	FMU, MMU ..
Model exporter	FMU exporter	Export model in a FMU format	FMU, tensorFlow Lite
Model exporter	**MMU exporter**	Export trained human model in an Motion Mock-up Unit. Could be a possible connection to WP3	MMU
	Co-Simulator	Simulation tool for executing FMUs	rfcspy, fmpy
	FAQ	Frequently asked questions	
	WIZARDS	A step-by-step tool for guiding users through processes.	

.

Table 2 List of AIToC AI platform modules

Parent module	Module	Description	Technologies	cardinality	Required in WP4
	Data importer	Application for import data from one format or system into another		1.. *	yes
Data importer	**Data structure parser**	Interpretation of e.g. AutomationML to determine the "path of the data"/"metadata"	AutomationML, Json, ...	1'	no
Data importer	File importer	Import data from a file	CSV, HDF5, ...	0.. *	yes
Data importer	AIToC-Data-Marketplace Conector	Import and Export data on Data-Marketplace	tbd	1'	yes
Data importer	**DB connector**	Import and export data to certain data bases	InfluxDB, MariaDB (SQL), timescaleDB, postgres (noSQL)	0.. *	no
Data importer	**Message broker connector**	Publish and subscribe on certain message broker	rabbitMQ, OrionLD, Kafka...	0.. *	no
Data importer	**Schema designer**	Configuration of the data structure	RDF, OWL	0.. 1	no
Data importer	**Data checker**	Proof imported data on the base of used schema	tbd	0.. *	no
Data importer	**Data backbone**	Import data, publish and subscribe data	Kafka	1'	no
	Data visualization	Data can be visualized in diagrams or tables	matplot, graphana, seaborn	0.. *	yes
	User Interface	Helps the user to select, configure, and perform settings on the platform for their pipelines.	Dear ImGui, NodeRed ...		
User Interface	Pipeline selector	Filter and select prepared pipeline and suggestion on base of the imported data	tbd	1'	yes
	Pipeline templates	Reusable pipelines	gRPC, Docker, Kubernetes, ARGO ...	0.. *	yes
pipeline templates	Preprocessor (Data-integration)	Reusable preprocessing pipelines	NumPy,Pandas	1.. *	yes
Preprocessor (Data-Integration)	Preprocessing methods (Data -cleaning/ -reduction/ -transformation)	Individual preprocessing methods	tbd	1.. *	yes
pipeline templates	Preprocessor (AI)	Reusable preprocessing for AI pipelines	tbd	1.. *	yes
Preprocessor (AI)	Preprocessing methods (AI specific)	Individual preprocessing for AI methods	tbd	1.. *	yes
pipeline templates	Training pipeline	Reusable AI pipelines	tbd	1.. *	yes
Training pipeline	AIM-Tool connector	Access to e.g. NN-Tool and transfer preprocessed data and import trained model to platform	tbd	1'	yes
Training pipeline	AIM-Designer	Configure, select and parametrize e.g. the neurenal networks	KERAS (Tensor Flow, Torch)	1'	yes
Training pipeline	AIM-Trainer	Execution of the training and validation of the trained model	KERAS (Tensor Flow, Torch)	1'	yes
	Model exporter	Export it in common simulation format	FMU, MMU ..	1.. *	yes
Model exporter	FMU exporter	Export model in a FMU format	FMU, tensorFlow Lite	0.. 1	yes
Model exporter	**MMU exporter**	Export trained human model in an Motion Mock-up Unit. Could be a possible connection to WP3	MMU	0.. 1	no
	Co-Simulator	Simulation tool for executing FMUs	rfcspy, fmpy	0.. 1	no
	FAQ	Frequently asked questions		0.. 1	no
	WIZARDS	A step-by-step tool for guiding users through processes.		0.. *	no

4.1.6. Step 5: Development cycle

The development cycle incl. component development will be part of the next phase of AIToC. The component development takes place in iterative cycles that continuously increase the maturity of the overall system. This includes the data provisioning, data preprocessing and machine learning component development dedicated to one of the present use cases and additionally in a generalized way where possible. The result will be a software prototype (Deliverable D4.2).

4.2. AITOC AI Methods and Model-Architectures

The data used in our use cases for the training of AI-models can be characterized as time series data. Furthermore, since we are trying to model production equipment/machines, the data not only describes behavior based on physical phenomena (e.g. readouts of various sensors measuring physical properties) but also various digital signals,who depend on the internal processing logic of the machine to be modeled.

This data is often highly cyclical, since machines often perform the same or similar tasks repeatedly.

Time series data can be noisy and have outliers, which can affect the model's performance. This challenge must be faced during data preprocessing phase to allow for good AI-model training results.

Time series data is often highly correlated, with the current value being dependent on the previous values. This means that the model needs to consider the temporal dependencies in the data.

A fundamental model architecture choice is the decision for a single step or multi-step approach (see Figure 21).

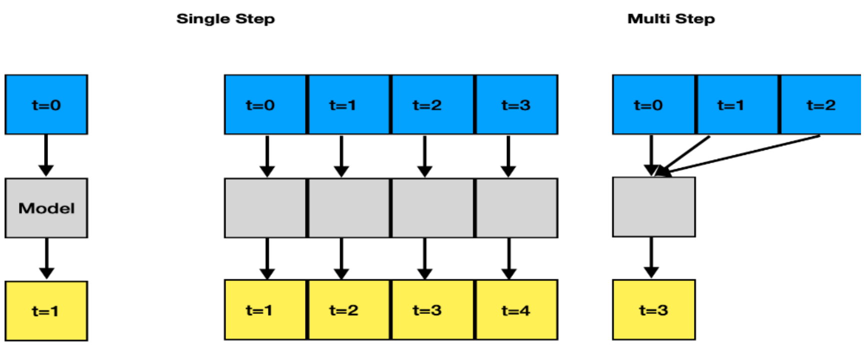

Figure 21 Difference between single step and multi-step approaches

In single step approaches, only one time step is used as input for the prediction. In multi-step approaches, a window of several time steps is used as input for the prediction model. The advantage of a multi-step approach is that the model has much more context for the input data during training and prediction. A disadvantage is that during inference a latency for the predicted values dependent on the input data window size is introduced.

Overall, classical machine learning approaches, such as linear regression or decision trees, and modern deep learning methods both have their strengths and limitations for time series prediction, and the choice between the two will depend on the specific characteristics of the time series data and the amount of available data for training. Deep learning methods tend to be more powerful and efficient for tasks with complex patterns and large amounts of data, while classical machine learning approaches and statistical methods may be more suitable for tasks with simpler patterns or smaller amounts of data.[3]

To address the mentioned challenges, various model architectures have been investigated, such as multi-layer perceptrons (MLPs), recurrent neural networks (RNNs) and especially long short-term memory (LSTM) networks.

LSTM networks (see Figure 22) are one variant of RNNs. RNNs and especially LSTM networks can capture the temporal dependencies in the data and save internal states. They have internal memory cells and gates that allow them to store and selectively expose information from the past, which makes them well-suited for tasks where the input data has complex temporal dependencies, such as in time series prediction. LSTM networks can require more computational resources and take longer to train, but they can potentially achieve better performance on tasks with complex temporal dependencies. [9]

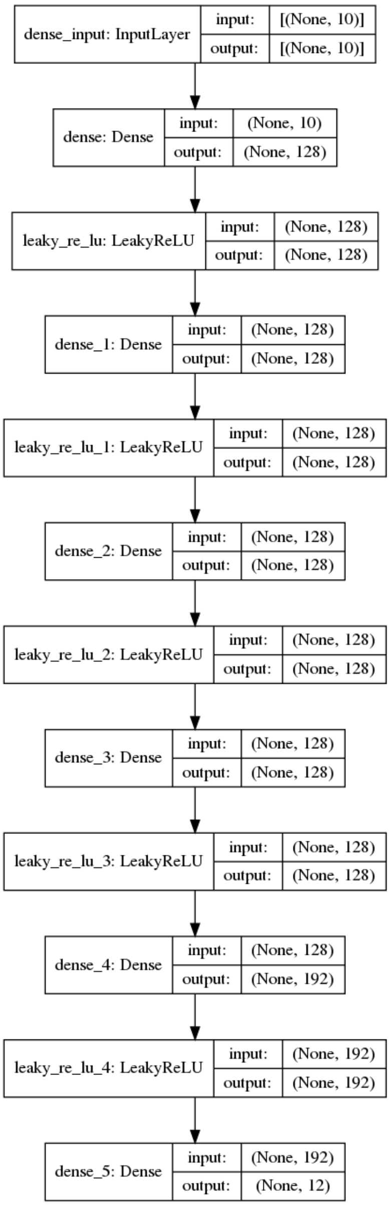

Figure 22 Example architecture of a long short-term memory (LSTM) network

To be able to use LSTM networks in an effective way, multiple model architecture variations need to be tested.

During the training phase, different sets of hyperparameters (e.g. the learning rate) have to be tried out to achieve the best possible model performance.

Variations will be further evaluated during implementation of the SW prototype.

First results are already promising (see Figure 23and Figure 24).

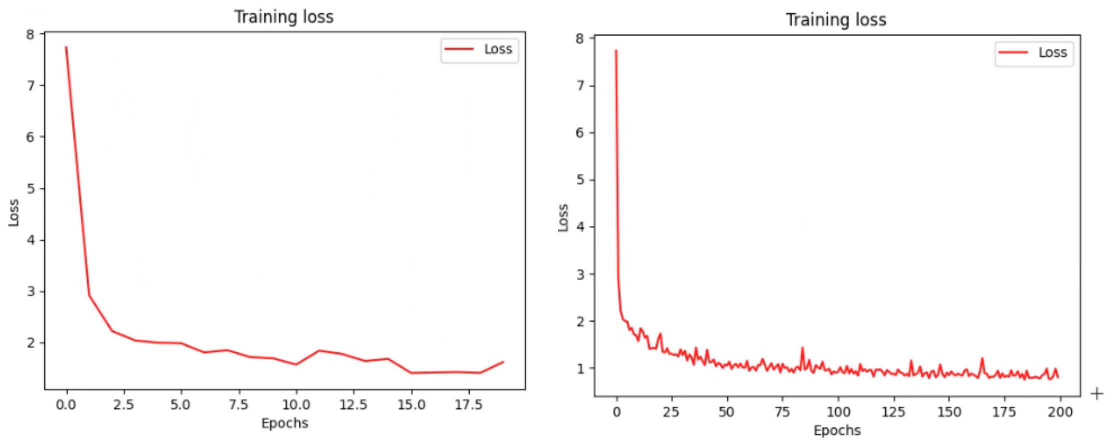

Figure 23 Example of the training loss of a LSTM prediction model for a frequency converter

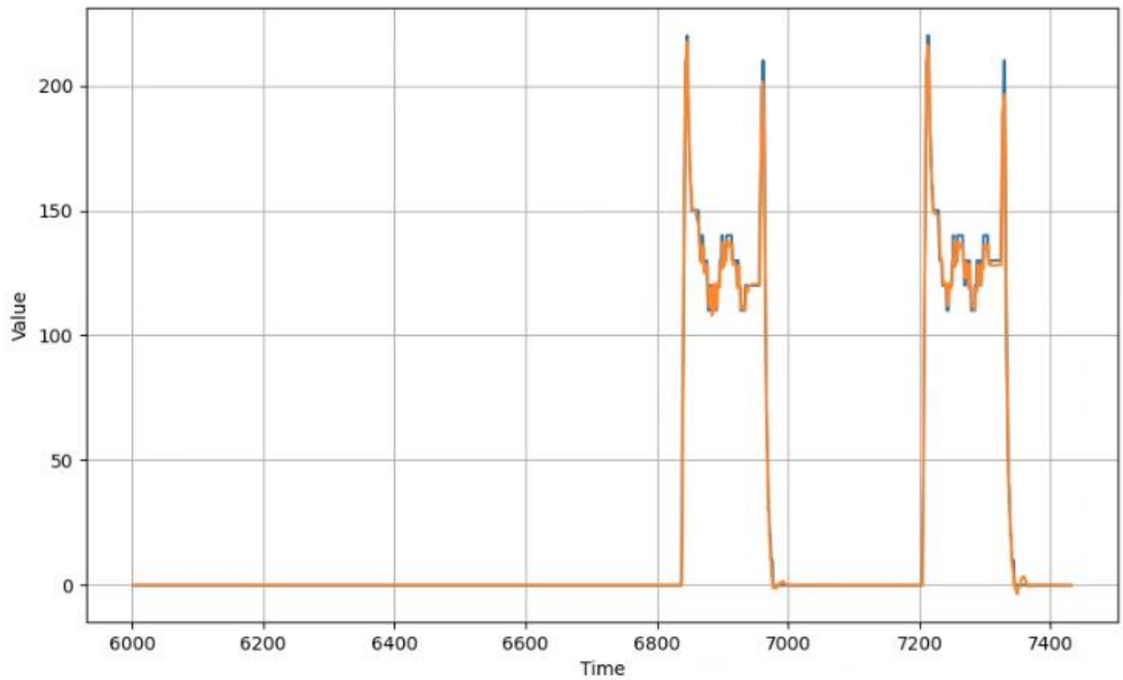

Figure 24 Comparison of prediction (orange) vs. ground truth (blue) of an analog sensor value of a frequency converter

5. Domain Storytelling

In software development, a user story is a brief, informal description of a feature or functionality that is desired by a user. It is typically written from the perspective of the user and describes the value that the user will receive by using the feature. In this work, the domain story of individual use cases was determined in several workshops. In this process, the roles, systems, and workflow of the individual use cases were built up in the domain story in the context of the platform to be implemented.

5.1. EKS/ISB Domain Story

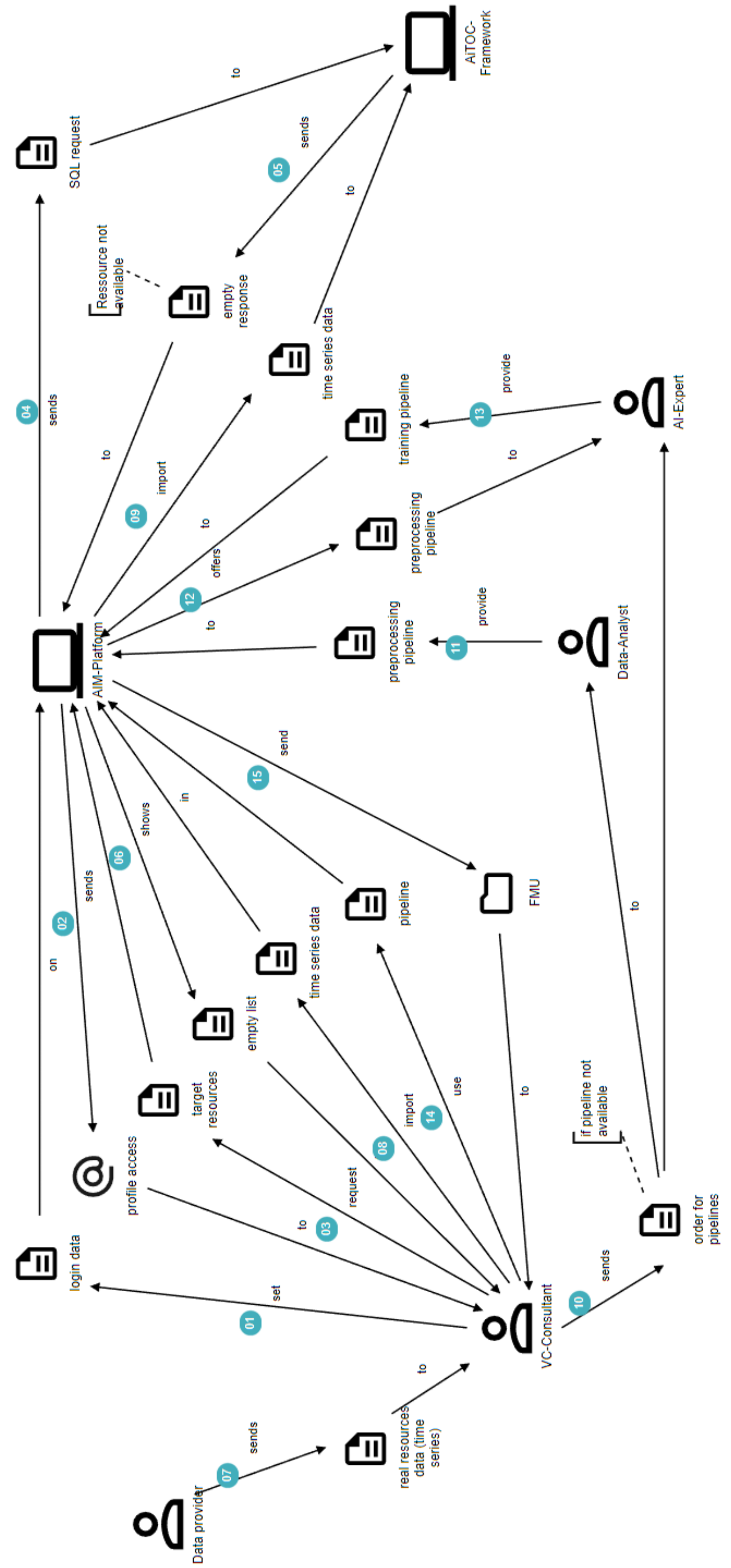

Figure 25 Domain story of a VC-Consultant in EKS InTec

ROLES:

Data-Provider (Customer):

The data provider is responsible for supplying the raw data. In this use case, the data is extracted from a real plant as time series data and labeled with the corresponding signal names. The provider can be either the component manufacturer as well as the plant operator.

EKS VC-Consultant:

The VC consultant is responsible for the preparation and execution of a virtual commissioning. To build the virtual model of the plant, behavior models of the components are needed. These often require a deep understanding of the components which requires a domain know-how that has to be built up with high effort.

ISB Data Analyst:

The Data Analyst is responsible for the building and parametrization of a processing pipeline, consisting of predefined AIM platform processing nodes, to convert the provided raw data into a simulation model. He may also implement new processing nodes, for example to implement new preprocessing functionality.

ISB AI-Expert:

The AI-Expert is responsible for the design and implementation of the Machine Learning Models and their integration in the AIM Platform architecture.

SYSTEMS:

AIM-Platform:

A platform with the capabilities to import data, preprocess the data, train AI models and export these models.

AIToC-Data-Management:

Use of data management to be implemented in WP2.

WORKFLOW: creating a behavior model of a frequency converter for the use in a virtual commissioning via the AIM-Platform

The following steps describe the relationships and the workflow between the roles and systems described above. This is demonstrated by the figure shown in the form of a domain story.

- The intention of the VC Consultant is the creation of a behavior model, which he can use for Virtual Commissioning. The model must be compliant with the requirements of the plant to be prepared and provided in FMU format
- In the first step, the VC consultant has to log in to the AIM-Platform
- VC-Consultant gets a profile access for AIM-Platform
- The AIM platform has a link to the AIToC data management, where the user has the ability to query data.
- If the necessary data is not available, an empty list is sent to the user as a response.
- In order to obtain suitable raw data, the component manufacturer or a plant manufacturer is consulted, who provides a set of suitable data.
- The received data is imported by the user to the AIM-Platform.
- The imported Data can be stored local or is imported into AIToC-Data-Management
- The next step is to check if the required predefined processing pipeline for the given input data (e.g. raw sensor data of a frequency converter) and the desired output data (e.g. a FMU model of the frequency converter) are available on the platform.
- If this is not the case, further experts must be involved.
- If no predefined processing pipeline is available, a Data-Analyst must construct and parametrize a suitable pipeline on the AIM Platform, consisting of individual predefined processing nodes. If needed (pre-)processing functionality is not already available as an implemented module, the data analyst will implement this module for the AIM Platform.
- If no suitable AI-module for the desired pipeline is available, the AI-Expert will design the machine learning model and then implement the respective AIM-Platform module.
- This process will usually also lead to a first complete processing pipeline template on the AIM-Platform, which can then be reused or adapted by other users.

- These pipelines can be run by the VC-Consultant on the AIM platform and the previously imported data.
- Afterwards, the trained model can be exported as an FMU.

5.2. ARD/Tofas Domain Story

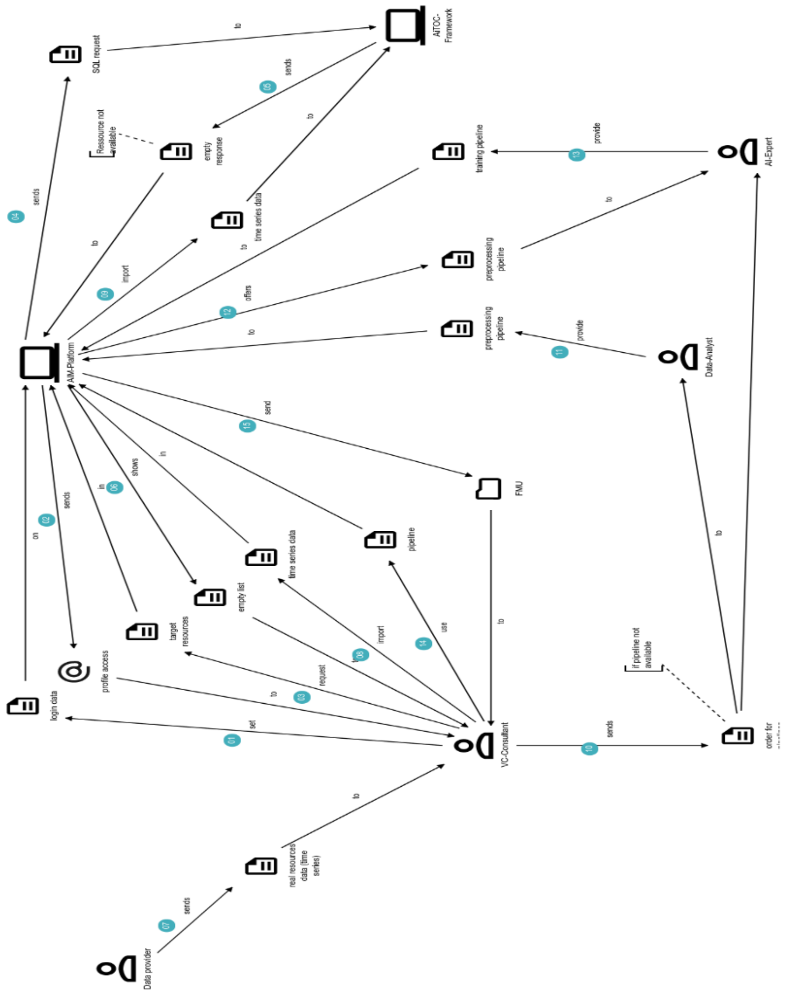

Figure 26 Domainstory from ARD/TOFAS

ROLES:

The roles can be assumed by a single person, or multiple persons with required skills. The roles we defined are; (i) Data-Provider, (ii) VC Consultant, (iii) Data Analyst, and (iv) AI-Expert. Definition of these roles are same as of the definition given in EKS domain.

SYSTEMS:

ARD AITOC Studio is being developed by ARDGroup to enable novice programmers to do the required task within a platform, in order to develop an AI Model. AITOC Studio provides different connectors, which connects and acquire data from different data sources, such as CSV file, Video File, MQTT broker, etc. It provides functionality to create projects containing one or more pipelines in it. Different pipelines can be created for different purposes, such as Pre-processing pipeline, ETL Pipeline, Training and Model Creation Pipeline, Testing Pipeline, etc. There is no limitation for the number and type of pipeline. That means user can create as many pipeline as needed. We are designing our solution in a way that, any pipeline can be saved as a single component, and be utilized in many projects.

WORKFLOW:

- Data Provider shares the data with VC-Consultant
- VC-Consultant
  - Enters into system,
  - Creates a project in workspace,
  - Uploads data and/or selects appropriate connector
  - Gives access to relevant persons
  - Sends order to data analyst to analyze the data and prepare relevant pipelines, such as pre-processing pipeline, visualization pipeline, etc.
- Data Analysts analyze the and creates pipeline to make it ready for model creations
- AI Expert accesses the system,
  - Checks the pipeline results
  - Uses the Visualization Pipeline to see the results
  - Creates further ETL pipelines if needed
  - Creates an AI Model, and trains it
  - Tests the resultant model(s) and compares the accuracy
- VC-Consultant exports the resulting model as a FMU , and tests it with unseen data

5.3. TWT Domain Story

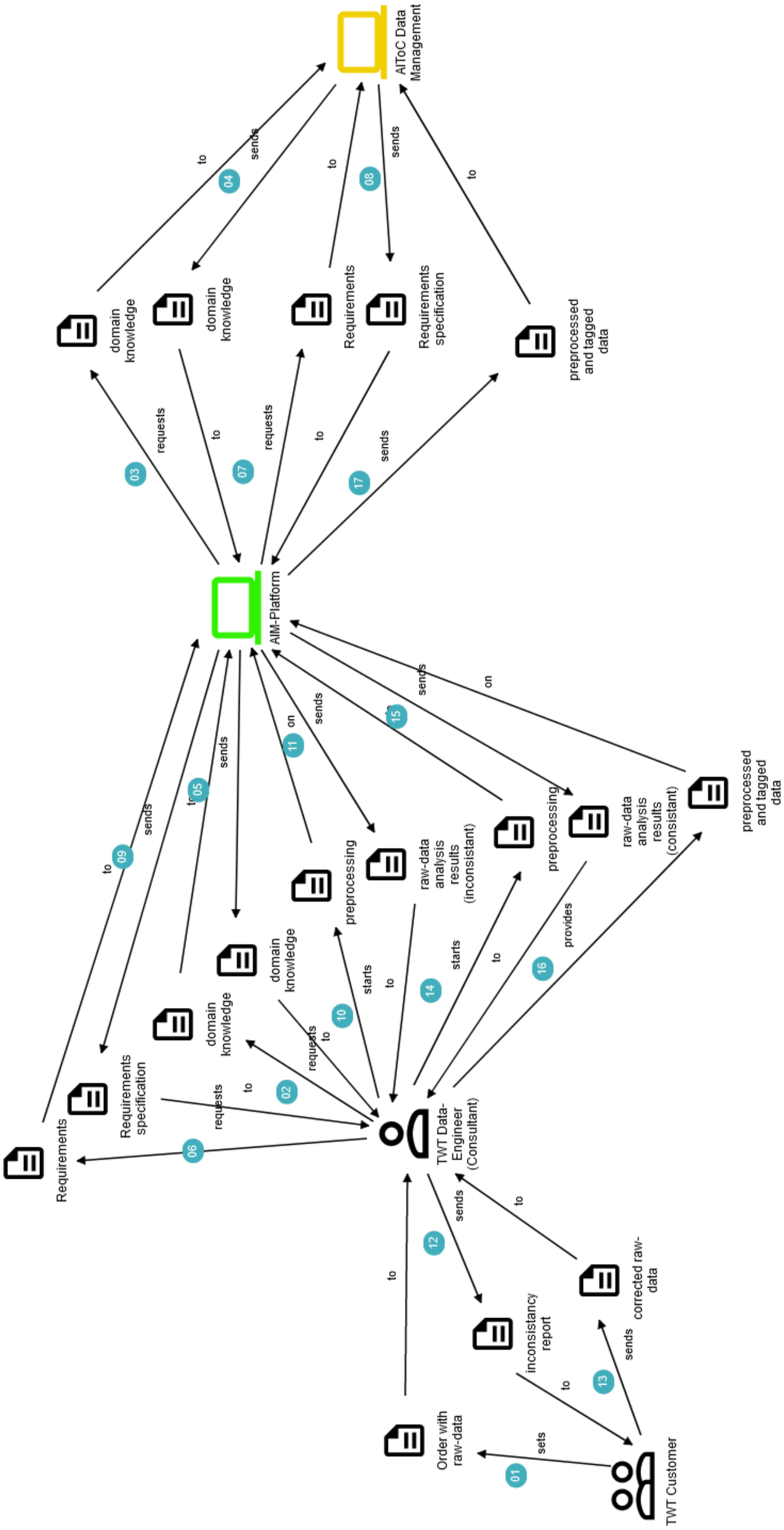

Figure 27 Domain story of a Data Engineer (consultant) at TWT

ROLES:

TWT Customer (data provider):

The TWT Customer orders several preprocessing steps for their raw data to prepare it for later usage. The preprocessing steps that the TWT Customer will select from are data cleaning, such as correcting inconsistent data, data integration, data reduction, data transformation, and/or data tagging/labeling. The TWT Customer provides the raw data, which often is some time series data extracted from a real system such as measured at a production facility.

TWT Data Engineer (Consultant):

The TWT Data Engineer is responsible for preprocessing the customer's raw data. This often requires an understanding not only of the data preprocessing, but also of the domain and the requirements to be addressed. The TWT Data Engineer must be able to select, to parametrize and to make use of the most suitable preprocessing pipeline on the AIM-Platform. Sometimes the TWT Data Engineer also has to tag or label the data.

SYSTEMS:

AIM-Platform:

A platform with the capabilities to import data, preprocess the data, train AI models, and export these models.

AIToC-Data-Management:

Solution for project-wide data management to be implemented in WP2.

WORKFLOW: preprocessing raw data via the AIM-Platform

The following workflow steps describe the interactions and dependencies between the roles and systems introduced above. This workflow is also depicted in the previous figure "Domain story of a Data Engineer (consultant) at TWT" and the preceding numbering corresponds to the numbering in that figure, see Figure 28.

- A TWT Customer orders several preprocessing steps for their raw data to prepare it for later usage. The TWT Customer also provides the raw data.
- (02&03) For preprocessing the customer's raw data the TWT Data Engineer needs the corresponding domain knowledge. The TWT Data Engineer

requests the corresponding domain knowledge from the AIM-Platform, which in turn request this domain knowledge from the AIToC-Data-Management.

- (04&05) The AIToC-Data-Management sends the domain knowledge, for example in the form of a domain ontology, to the AIM-Platform, which in turn makes it available to the TWT Data Engineer.
- (06&07) For preprocessing the customer's raw data the TWT Data Engineer also needs the requirements that have to be addressed. The TWT Data Engineer requests the corresponding requirements from the AIM-Platform, which in turn request these from the AIToC-Data-Management.
- (08&09) The AIToC-Data-Management sends the requirements to the AIM-Platform, which in turn makes them available to the TWT Data Engineer.
- (10) For the available raw data the TWT Data Engineer selects, parametrizes and makes use of the most suitable preprocessing pipeline on the AIM-Platform. The performed preprocessing steps can be data cleaning, such as correcting inconsistent data, data integration, data reduction, data transformation, and/or data tagging/labeling. The TWT Data Engineer starts the preprocessing of the raw data on the AIM-Platform.
- (11) After preprocessing the raw data the AIM-Platform provides the TWT Data Engineer with some raw data analysis results.
- (12to15) If the raw data analysis results of the preprocessing performed are inconsistent, the TWT Data Engineer contacts the TWT Customer with an inconsistency report and requests a correction of the raw data. The TWT Customer provides corrected raw data, and the TWT Data Engineer repeats the preprocessing of the new raw data on the AIM-Platform.
- (16&17) If the raw data analysis results of the preprocessing performed are consistent, the TWT Data Engineer provides the preprocessed and, if needed, tagged data to the AIM-Platform, which in turn stores this data in the AIToC-Data-Management. This is the final step of this workflow.

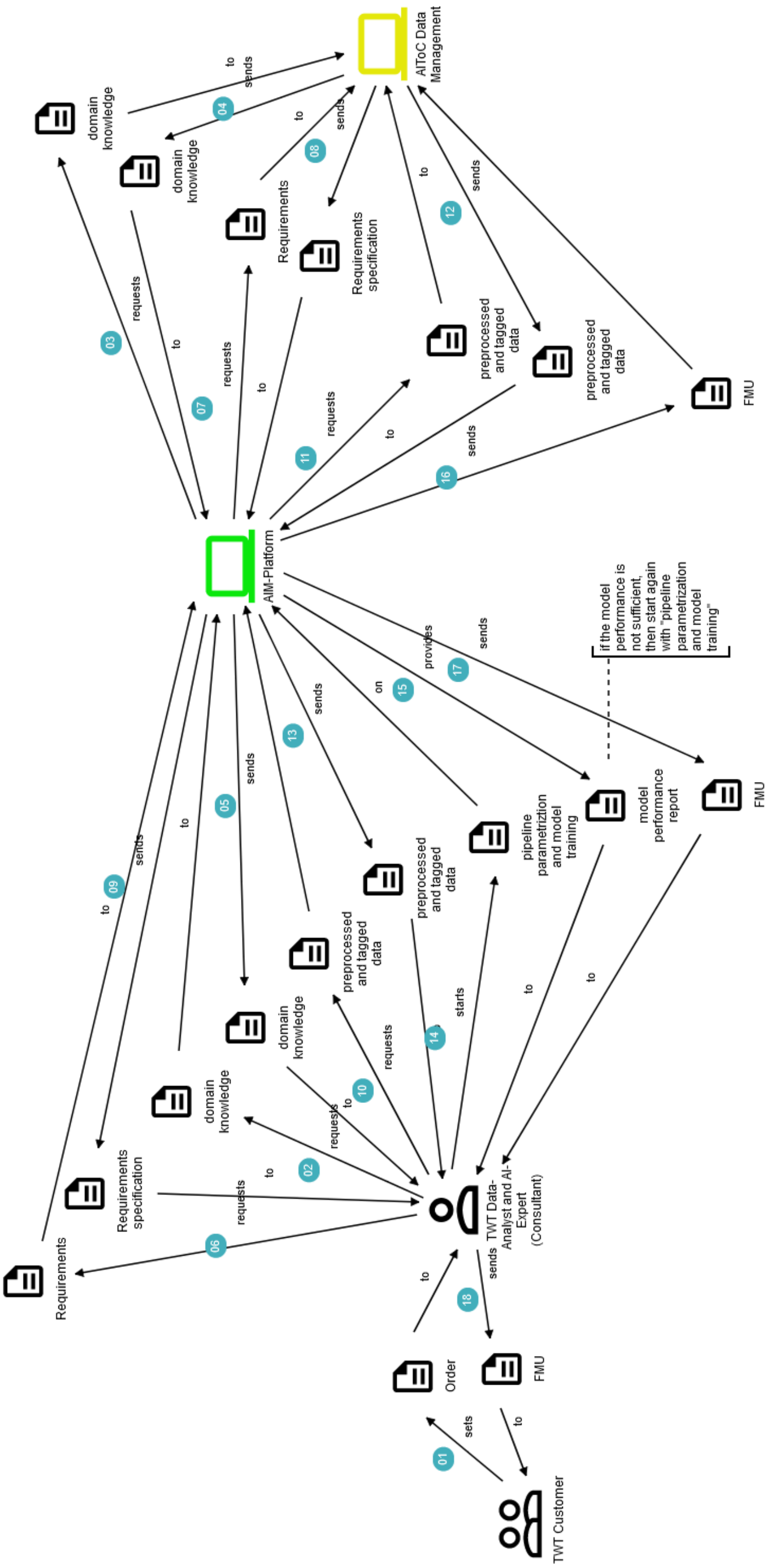

Figure 28 Domain story of a Data Analyst and AI-Expert (consultant) at TWT

ROLES:

TWT Customer (data provider):

The TWT Customer orders a behavior model that is created based on preprocessed data and provided in the FMU format. This behavior model must satisfy some corresponding requirements.

TWT Data Analyst and AI-Expert (Consultant):

The TWT Data Analyst and AI-Expert is responsible for the creation of a behavior model out of preprocessed data. This often requires a deep understanding not only of the analysis and training approaches and methods, but also of the domain and the requirements to be addressed. The TWT Data Analyst and AI-Expert must be able to select, to parametrize and to make use of the most suitable training pipeline and the most suitable AI-model on the AIM-Platform.

SYSTEMS:

AIM-Platform:

A platform with the capabilities to import data, preprocess the data, train AI models and export these models.

AIToC-Data-Management:

Solution for project-wide data management to be implemented in WP2.

WORKFLOW: creating a behavior model wrapped in the FMU format based on preprocessed data via the AIM-Platform

The following workflow steps describe the interactions and dependencies between the roles and systems introduced above. This workflow is also depicted in the previous figure “Domain story of a Data Analyst and AI-Expert (consultant) at TWT” and the preceding numbering corresponds to the numbering in that figure.

- A TWT Customer orders a behavior model that shall be created based on preprocessed data and provided in the FMU format.
- (02&03) For the creation of a behavior model the TWT Data Analyst and AI-Expert needs the corresponding domain knowledge. The TWT Data Analyst and AI-Expert requests the corresponding domain knowledge from the AIM-Platform, which in turn request this domain knowledge from the AIToC-Data-Management.

- (04&05) The AIToC-Data-Management sends the domain knowledge, for example in the form of a domain ontology, to the AIM-Platform, which in turn makes it available to the TWT Data Analyst and AI-Expert.
- (06&07) For the creation of a behavior model the TWT Data Analyst and AI-Expert also needs the requirements that have to be addressed. The TWT Data Analyst and AI-Expert requests the corresponding requirements from the AIM-Platform, which in turn request these from the AIToC-Data-Management.
- (08&09) The AIToC-Data-Management sends the requirements to the AIM-Platform, which in turn makes them available to the TWT Data Analyst and AI-Expert.
- (10&11) For the creation of a behavior model the TWT Data Analyst and AI-Expert also needs the preprocessed and tagged data. The TWT Data Analyst and AI-Expert requests the corresponding data from the AIM-Platform, which in turn request the data from the AIToC-Data-Management.
- (12&13) The AIToC-Data-Management sends the preprocessed and tagged data to the AIM-Platform, which in turn makes it available to TWT Data Analyst and AI-Expert.
- (14) For the available preprocessed data the TWT Data Analyst and AI-Expert selects, parametrizes and makes use of the most suitable training pipeline and the most suitable AI-model on the AIM-Platform. The TWT Data Analyst and AI-Expert parametrizes the pipeline and starts the model training on the AIM-Platform.
- (15&14) After training the model the AIM-Platform provides the TWT Data Analyst and AI-Expert with a model performance report. If the model performance is not sufficient, the TWT Data Analyst and AI-Expert adapts the pipeline parametrization, repeats the model training on the AIM-Platform, and evaluates the new model performance. This is repeated until the model performance is sufficient.
- (16) If the model performance is sufficient, the behavior model is finalized, wrapped as a FMU and stored by the AIM-Platform in the AIToC-Data-Management.
- (17&18) The TWT Data Analyst and AI-Expert receives the finalized FMU from the AIM-Platform and provides it to the TWT Customer. This is the final step of this workflow.

5.4. LUT/Tactotec/Process Genius Domain Story

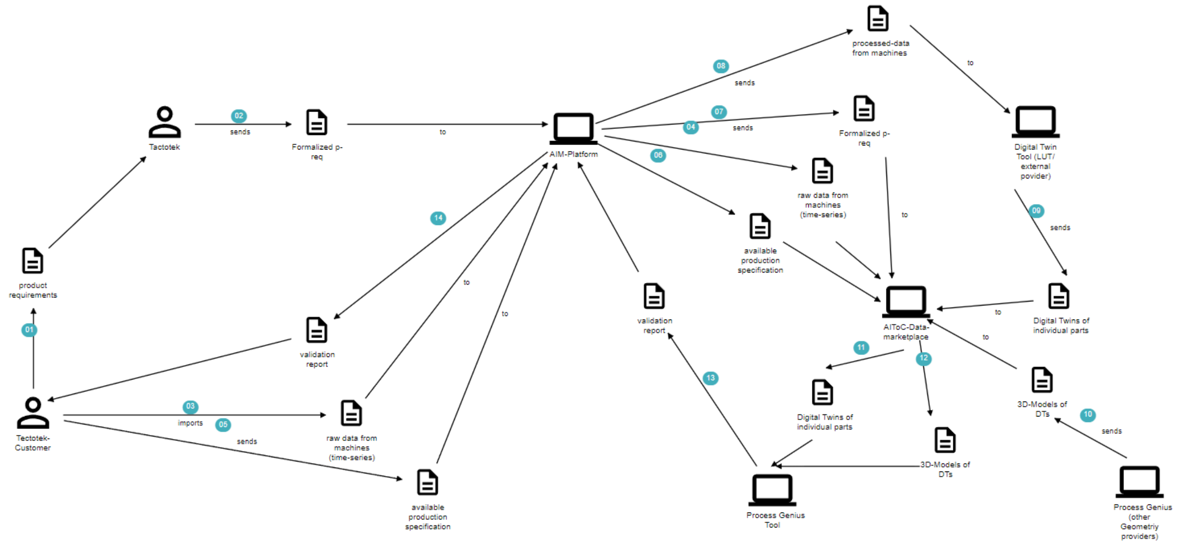

Figure 29 Domain story of a Data Engineer (consultant) at LUT/Tactotec

ROLES:

Customer – end user for the service and buyer of the Tactotek's service.

Salesperson – receives orders from customers and forms relations with customers, providing also basic support.

Production engineer – manufacturing expert, provides know how and technical support.

Researcher – provides algorithms and models for the system elements.

3D modeller – provides 3D models of the production equipment that can be used in the simulation.

SYSTEMS:

AIM-platform – used as simulation platform and digital twin training tool.

Process Genius tool – web user interface providing convenient production line modelling tool and result visualization platform.

AIToC data marketplace – data storage and management platform serving as back end for the AIM platform.

WORKFLOW:

Customer contacts Tactotek to request offer for a specific production line. Initial specification of requirements is provided in non-standard format, being just human readable but not formalized. Salesperson converts informal requirements into formalized requirements as defined in WP1. Formalized requirements are sent to the AIM platform that is configured with Tactotek's production process specific models. AIM platform generates validation report for the requirements. In the meantime, salesperson creates a system account for the customer, enabling customer to directly fetch validation report from the AIM platform. Customer can then supply AIM platform with the raw data from production machines that he has at his disposal and that he would like to reuse in the new product manufacturing process. He also supplies available production specification and can modify requirements using formalized requirements definition tool. All the data is transferred to the AIToC data marketplace and is assigned to specific project and customer. AIM platform is used to generate parametrization of digital Twins for customer's manufacturing equipment. In addition, based on documentation 3D modeler at Process Genius develops 3D models of the machines for visualization purposes in the Process Genius tool. Digital twin models for Tactotek's production process equipment are defined at LUT and provided as parametric models to the AIToC data marketplace, through which are made available to the AIM platform.

Customer can use Process Genius user interface tool to configure production line based on available equipment and other suggested equipment available in the back end's library. The simulation is then executed on the AIM pipeline and results are collected through data marketplace. Then those results are available in the Process Genius user interface tool for visualization of utilization of individual machines, process costs, setup costs, throughput of the machines and the line and other process related information. Once customer is satisfied with the results, he places an order in Tactotek for technology license and is able to further optimize production plant design and operation parameters utilizing support from production engineers.

6. User Interface

6.1. Concept

The user interface is based on the concept of a node editor.

A node editor is a type of user interface that allows users to create and edit data pipelines by connecting different processing nodes together.

The nodes represent individual processing modules, and the connections between the nodes represent the flow of data through the pipeline.

In the node editor, each processing module is represented by a box that contains information about its function, its status and configuration parameters, its input and output data.

Users can create new nodes by clicking on a button or menu option, and can delete existing nodes by clicking on them and pressing the delete key.

The connections between the nodes are represented by edges that show the flow of data. Users can create new connections by clicking on the output port of one node and dragging it to the input port of another node.

The node editor also has a library pane that contains pre-built nodes and pipeline templates.

In addition to the basic functionality described above, the editor also has special data visualization nodes that can be inserted into the pipeline. These nodes allow the user to view the data at that point in the pipeline using various visualization tools, such as plots, charts, and tables. By including data visualization functionality, the user will be able to more effectively understand and work with the data as it flows through the pipeline, and make data-driven decisions about how to optimize and improve the pipeline.

6.2. Design

Key elements of the UI, see Figure 30:

- A canvas area allowing for the construction of the data pipeline
- Nodes, representing the individual processing modules of the pipeline. Each node has UI elements for the configuration of the relevant parameters of the underlying processing module
- Edges, representing the data flow from one processing module to the next.
- Data visualization windows
- A library pane that allows for the selection of processing nodes and whole pipeline templates and the saving of configured pipelines
- A pane allowing for the execution control of the processing pipeline

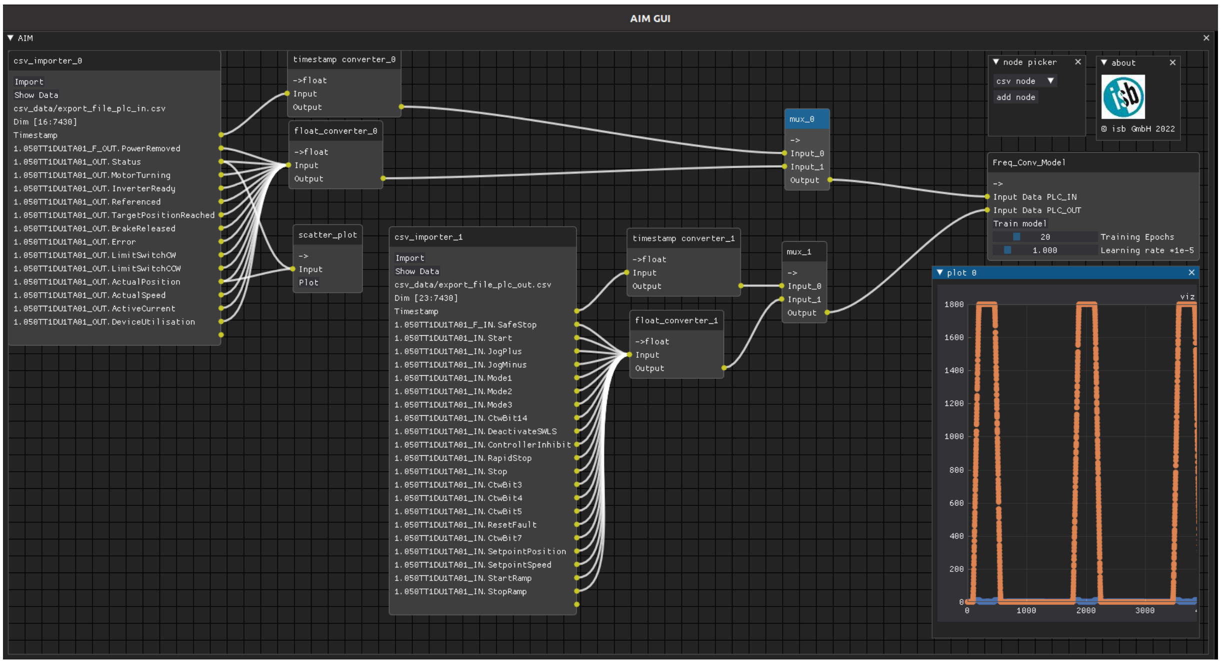

Figure 30 UI Prototype demonstrating the key functionalities of the user interface

7. Summary and conclusions

In this Task, we have taken a closer look at the data of various use cases and explored the relationship between these use cases and the Data Market from Work Package 2. Our main focus has been on the GRM workflow, which closely ties in with the results of the Data Market. This allows us to extract and use data from the market as well as export the processed data back into it.

Moving on, we have also investigated the preprocessing of data in the PRM workflow. This has resulted in the creation of a decision tree, which provides the user with a clear path to choose the most suitable preprocessing methods based on the available data. In addition, we have delved into the utilization of AI methods in engineering by means of the PAISE model.

To round off the work, we have conceptually depicted the usage of the platform by different partners through Domain Stories. Lastly, we have presented initial ideas for the user interface in the final section of the work, providing a glimpse into how the platform will be used by the end-users.

8. Bibliography

[1] Deniz Neufeld. Visualization methods for periodic time series. Cognitive Systems Group, Otto-Friedrich University Bamberg, An der Weberei 5, 96047 Bamberg, Germany, 9, 03 2021.

[2] Cheng Fan, Meiling Chen, Xinghua Wang, Jiayuan Wang, and Bufu Huang. A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Frontiers in Energy Research, 9, 03 2021.

[3] Hongzhi Wang, Mohamed Jaward Bah, and Mohamed Hammad. Progress in outlier detection techniques: A survey. IEEE Access, 7:107964–108000, 2019.

[4] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. ACM computing surveys (CSUR), 50(6):1–45, 2017.

[5] Sagar S Badhiye and PN Chatur. A review on time series dimensionality reduction. HELIX, 8(5):3957–3960, 2018.

[6] CC-KING: PAISE® The process model for AI systems engineering, https://www.ki-engineering.eu/content/dam/iosb/ki-engineering/downloads/PAISE(R)_Whitepaper_english.pdf (accessed 21.12.2022)

[7] DIN/DKE: DEUTSCHE NORMUNGSROADMAP KÜNSTLICHE INTELLIGENZ DEUTSCHE NORMUNGSROADMAP KÜNSTLICHE INTELLIGENZ AUSGABE, 2, https://www.din.de/resource/blob/891106/57b7d46a1d2514a183a6ad2de89782ab/deutsche-normungsroadmap-kuenstliche-intelligenz-ausgabe-2—data.pdf, 2022.

[8] Hewamalage, Hansika, Christoph Bergmeir, and Kasun Bandara. "Recurrent neural networks for time series forecasting: Current status and future directions." International Journal of Forecasting 37.1 (2021): 388-427.

[9] CHANDRA, Rohitash; GOYAL, Shaurya; GUPTA, Rishabh. Evaluation of deep learning models for multi-step ahead time series prediction. IEEE Access, 2021, 9. Jg., S. 83105-83123.