PADERBORN UNIVERSITY
The University for the Information Society

Project Group:

Machine Learning for Predictive Maintenance

(Weekly Status Report)

Supervisors:
Prof. Dr. Eyke Hüllermeier (eyke@upb.de)
Tanja Tornede (tanja.tornede@upb.de)

Name: Paul Fährmann
Matriculation Number: 7006813
E-mail: pauldf@mail.uni-paderborn.de

Summary of main contributions

Introduction to PdM

1. Created GitLab structure and folder structures for our documentation repository

2. Connected GitLab update information of both repositories (code and documentation) with Microsoft Teams (with Christopher)

Topic Study & Requirement Analysis

1. Planning the distribution and topics to include from the survey papers in the Time Series Feature Extraction (TSFE) topic (with Sanjay). Also further discussions about how to extend our chapter later on.

2. Discussed, created explanations and formulated the formal definition for time series.

3. Searched and added quotes for the different chapters.

4. Wrote the Formal Definition, Frequency Domain Approaches (DFT for basic features, BOSS), Time and Frequency Domain Approaches (DWT, EMD) and the Evaluation Setup in the TSFE chapter. Also created a lot of new, clean visualizations for these approaches.

5. Reviewed Introduction, Conclusion, References, Health State and RUL chapters

System Design

1. Was part of the discussion and creation the PDMFF file format.

2. Also discussed the abstract class diagrams with Sanjay and Christopher, including conventions about alignment, naming, inheritance and other details. Aligned the final version of our abstract class diagram.

3. Discussed changes to the loops of the general sequence diagram.

4. Created sequence diagrams for PytsTransformWrapper, TimeSeriesImputer, PywtWrapper, EMDSignalWrapper, UniToMultivariateWrapper and added explanations of them in our system design document.

5. Created a testing strategy based on *pytest* with three levels: Unit, Integration and System.

6. Developed the package folder structure of our code repository.

Implementation

1. Created our "code" repository in GitLab including package structure, master and develop branch, gitignore and included our merge request template.

2. Implemented the following classes completely: Evaluator, UniToMultivariateWrapper, PytsTransformWrapper, PytsSupportedAlgorithm, EMDSignalWrapper, PywtWrapper, TimeSeriesImputer, AttributeFilter, DatasetToSklearn and ML4PdM.

3. Also included multiprocessing capability for the UniToMultivariateWrapper.

4. Implemented the AttributeTypes system with their different classes and parser for NUMERIC, SETOFLABELS, DATETIME, TIMESERIES and MULTIDIMENSIONAL

5. Implemented several useful methods in other classes including generate_simple_cut_dataset and get_time_series_data_as_array in Dataset and listify_time_series and attach_timesteps as basic functions.

6. Discussed a lot of issues about implementation and test coverage of the DatasetParser with Vinay.

7. Lots of personal discussions and reviews of implementations of classes for Sanjay and Vinay. Reviewed lots of official merge requests of all the other members, especially at the end of the implementation phase.

Final Presentation

1. Organized a lot of meetings with the group but also individual members to plan and distribute workload. Also started and lead the conversations around the motivation and "story" parts of the presentation.

2. Created the Challenges and Solutions Part int the Introduction part of the final presentation.

3. Created and described the How-To-Use example and our data format PDMFF in our documentation.

Calendar Week: 38 24 September, 2021

Completed Tasks

1. Changing the Example documentation according to discussion: Jupyter Notebook to be able to execute code snippets + simple .py code for easier access + links to the other pipeline notebooks.
2. Describing our PDMFF dataformat in a section of the documentation.

Tasks in-progress

1. Finishing the slides of challenges and solutions for the final presentation

Calendar Week: 37 17 September, 2021

Completed Tasks

1. finishing the output and config file version of the sipmle pipeline example.
2. discussed the better presentation slides and motivation (with team).
3. discussed the motivation and introduction slides with Vinay

Tasks in-progress

1. Changing the Example documentation according to discussion: Jupyter Notebook to be able to execute code snippets + simple .py code for easier access + links to the other pipeline notebooks.
2. Describing our PDMFF dataformat in a section of the documentation.

Calendar Week: 36 10 September, 2021

Completed Tasks

1. Discussed the distributed the tasks for the presentation as well as the final decision on the logo (with team).

2. Described a simple pipeline example in the documentation.

Tasks in-progress

1. finishing the output and config file version of the sipmle pipeline example

2. Describing our PDMFF dataformat in a section of the documentation.

Calendar Week: 35 03 September, 2021

Tasks in-progress

1. Writing and describing examples for the documentation on how to use our code to build pipelines and train them on data with all preprocessing and postprocessing steps.
2. Describing our PDMFF dataformat in a section of the documentation.

Calendar Week: 34 27 August, 2021

Completed Tasks

1. Helped Sanjay and Vinay with their implementation tasks and reviewed their merge requests.
2. Implemented RUL target recalculations for the WindowingApproach class
3. Fixed several issues regarding my implementations and document strings.

Tasks in-progress

1. Discussing and distributing the tasks of the presentation milestone

Calendar Week: 33 20 August, 2021

Completed Tasks

1. Developed simple cut dataset generator in the Dataset class which cuts ”full time series data” into a RUL regression task.

2. Developed multiprocessing capability for the UniToMultivariateWrapper, which then can process/transform multiple TIMESERIES attributes in parallel.

Tasks in-progress

1. Tests of the TsfreshWrapper and its functionality in pipelines on data. (with Sanjay)

2. Check all documentation strings of sphinx to be in the correct format in the documentation website that gets created automatically.

Calendar Week: 32 13 August, 2021

Completed Tasks

1. Developed ML4PdM wrapper and DatasetToSklearn Transformer.
2. Developed AttributeFilter transformer which omits certain attributes completely from a dataset. This can be directly indexed and/or conditioned on the number of unique values found in that attribute over the whole dataset.
3. Writing some integration tests with pipelines of multiple ml4pdm transformers and sklearn RandomForestRegressor. Results are not unreasonable.

Tasks in-progress

1. Further integration and system tests.

Calendar Week: 31 06 August, 2021

Completed Tasks

1. Finished PywtWrapper with it now using Dataset as input and output types.

2. Changing the input and output types of all my other transformer classes to only allow objects of type Dataset to be passed. These are EMDSignalWrapper, PytsTransformWrapper and PytsImputer.

3. Changed Evaluator to now use Dataset object as input and output.

4. Fixing the UniToMultivariateWrapper to streamline the functionality of putting the transformed and not transformed features back together. Also changing the output format of TimeseriesTransformer Subclasses to match the change.

Tasks in-progress

1. Developing ML4PdM wrapper and DatasetToSklearn Transformer.

2. Writing system tests with pipelines of multiple ml4pdm transformers and sklearn predictors. Also running synthetic and real cmapss data for those tests.

Calendar Week: 30 30 July, 2021

Completed Tasks

1. Finished and merged the AttributeType class and subclasses NUMERIC, DATETIME, TIMESERIES, MULTIDIMENSIONAL and SETOFLABELS to the develop branch. That included changes to the Dataset and Parser.

Tasks in-progress

1. Fixing the UniToMultivariateWrapper to streamline the functionality of putting the transformed and not transformed features back together. Also changing the output format of TimeseriesTransformer Subclasses to match the change.

2. Changing the input and output types of all my transformer classes to only allow objects of type Dataset to be passed.

Calendar Week: 29 23 July, 2021

Completed Tasks

1. Finished and merged the extension of the EMD Signal wrapper, which included the option of multi index output and filling the missing indices with 0 filled time series.

2. finished the code functionality of the PywtWrapper.

3. added a branch for the extension and fix to the Dataset class. Also added my getter method of the data with timeseries attributes transformed into simple lists.

Tasks in-progress

1. Testing and documenting the PywtWrapper.

2. Fixing the UniToMultivariateWrapper to streamline the functionality of putting the transformed and not transformed features back together. Also changing the output format of TimeseriesTransformer Subclasses to match the change.

3. Changing the input and output types of all my transformer classes to only allow objects of type Dataset to be passed.

Calendar Week: 28 16 July, 2021

Completed Tasks

1. Finished and merged the first version of the EMD-Signal Wrapper with test coverage and documentation.

Tasks in-progress

1. Extending the EMD Signal Wrapper to include the option of multi index output and filling the missing indices with 0 filled time series.
2. Implementing the PyWaveletWrapper with the 1D discrete wavelet transformation algorithm and checking what other options might be viable in this library.

Calendar Week: 27 09 July, 2021

Completed Tasks

1. Decided on the options for the EMD-Signal Wrapper on what components to keep.

Tasks in-progress

1. Finishing the EMD-Signal Wrapper test coverage
2. Testing of the Pywavelet library.
3. Writing the synthetic time series data set generator with different options

Calendar Week: 26 02 July, 2021

Completed Tasks

1. finalized the merge of the following classes: PytsTransformWrapper, TimeSeriesImputer with their documentation and test cases.

2. Tested the EMD-Signal library.

3. Wrote a listify function that changes the time series format from our representation to a representation that can be used in several different libraries (e.g. EMD-Signal and Pyts).

Challenges

1. Finding good ways and options on how to equalize the transformations of the different time series instances in such a way that the EMD-Signal wrapper can be used in a pipeline.

Tasks in-progress

1. Finishing the EMD-Signal Wrapper with options on how and what IMF components and/or the original time series to keep.

2. Testing of the Pywavelet library.

3. Writing the synthetic time series data set generator with different options

Calendar Week: 25 25 June, 2021

Completed Tasks

1. finished the documentation, test cases, module modifications of both classes PytsTransformWrapper and TimeSeriesImputer. Also created merge request for them.

Tasks in-progress

1. Finalizing the merge of the two classes mentioned above

2. Checking EMD Signal library and Pywavelet library for necessary formatting of our time series representation.

3. Writing the synthetic time series data set generator with different options

Calendar Week: 24 18 June, 2021

Tasks in-progress

1. Add automatic testing of different length time series to allow only certain Pyts algorithms for the PytsTransformWrapper.

2. Writing the testing methods and class for PytsTransformWrapper class specifically if and what Imputer algorithm works best for what pyts transformation algorithm.

3. Writing the testing methods and class for the TimeseriesImputer class.

Calendar Week: 23 11 June, 2021

Completed Tasks

1. Implemented functional TimeseriesImputer including the equalization of the lengths of the instances, and imputing the missing values by 2 handcrafted algorithms and by using the pyts InterpolationImputer class via keywords.

Tasks in-progress

1. Writing the testing methods and class for PytsTransformWrapper class specifically if and what Imputer algorithm works best for what pyts transformation algorithm.

2. Writing the testing methods and class for the TimeseriesImputer class.

Calendar Week: 22 04 June, 2021

Completed Tasks

1. Implemented base format changing code for the TimeSeriesImputer class including length equalizing and filling the resulting missing values with NaNs. This can then be used to apply any Imputer algorithm and get instances of equal lengths back.

2. Fixed some errors in the base format changing code for the PytsTransformWrapper for two different formats of time series. One is a list of pairs of (timestep, value), the other a simple list of values.

Tasks in-progress

1. Implementing PytsTransformWrapper with the enum design.

2. Implementing in tandem to the first task the TimeseriesImputer class.

Calendar Week: 21 28 May, 2021

Completed Tasks

1. Finished the transformation method, that transforms our time series representation into the representation that works with the classes of the pyts library.
2. Discussed code design ideas for the PytsTransformWrapper with Tanja including the usage of custom made enums in python.

Challenges

1. Facing an issue with a missing library for the pyts library to work properly. Needed to install the library ”numba” and needed to reinstall the whole ml4pdm anaconda environment in the process.

Tasks in-progress

1. Implementing PytsTransformWrapper with the enum design.

Calendar Week: 20 20 May, 2021

Completed Tasks

1. Finished UniToMultivariateWrapper and FixedSizeFeatureExtractor code, test coverage and documentation and merged it onto the develop branch.
2. discussed test coverage of the Dataset Parser with Vinay.

Challenges

1. Facing an issues with circular imports for which I needed to disable the autoformatting of imports in a module __init__.py file to fix it.

Tasks in-progress

1. Implementing PytsTransformWrapper

Calendar Week: 19 14 May, 2021

Completed Tasks

1. Finished evaluator code, test coverage and documentation and merged it onto the develop branch.

Tasks in-progress

1. Testing UniToMultivariateWrapper class with mock objects.
2. Finishing abstract classes

Calendar Week: 18 07 May, 2021

Completed Tasks

1. Finished first version code for UniToMultivariateWrapper, specifically the aggregation for fixed size feature extraction and time series transformation.
2. Discussed the current version of the dataset parser and representation of time series in the Dataset class with Vinay.

Tasks in-progress

1. Testing UniToMultivariateWrapper class with mock objects.
2. Changing the abstract classes with @abstractmethod and pass as discussed with the group.

Calendar Week: 17 30 April, 2021

Completed Tasks

1. Implemented rough/untested version of UniToMultivariateWrapper for wrapping univariate transformer classes for multivariate timeseries instances.

2. Researched Multi- and Univariate Timeseries Imputers including GAN, Recurrent Neural Networks and basic random sampling from the unimputed timeseries.

Tasks in-progress

1. Finishing UniToMultivariateWrapper class with tests.

2. Testing that synthetic and real datasets are running through the current system of Evaluator and Transformers/Regressors correctly.

Calendar Week: 16 23 April, 2021

Completed Tasks

1. Splitting up the rest of the abstract classes and functions/methods to implement from the general class diagram with the team.

2. Discussed changes to the Dataset Parser and Dataset classes with the team including static methods and what return value we expect.

Tasks in-progress

1. Implementing the FixedSizeFeatureExtractor class in a feature branch.

2. Checking the functionality of dataset class together with the evaluator class.

Calendar Week: 15 16 April, 2021

Completed Tasks

1. Finished the Evaluator class with current status of the other classes.
2. Created first test cases for that class with datasets from sklearn.datasets

Tasks in-progress

1. Waiting for Dataset class to finish to check the functionality with the Evaluator.
2. Checking ideas on how to ”fix” time series such that they can be used in the different libraries.

Calendar Week: 14 09 April, 2021

Completed Tasks

1. Created first code for the "Evaluator" class.

2. Added __init__ python files to mark the folder structure as a python module. Also added an import to be able to import a class directly from a directory.

3. Created test class for the "Evaluator" class.

Challenges

1. Setting up vscode with the correct python interpreter from the anaconda environment created a lot of problems that took several hours distributed over 2-3 days to fix with multiple reinstallations of the vscode IDE, Anaconda, the Anaconda environment as well as python itself.

2. There are a lot of cross information on how the classes are set up that are necessary to finish the code of these classes. That takes more time and iterations than expected.

Tasks in-progress

1. Filling in the rest of the missing code for the "Evaluator" as well as for the test class of the "Evaluator"

Calendar Week: 13 02 April, 2021

Completed Tasks

1. Set up Visual Studio Code IDE with Python extension.
2. Set up anaconda environment via anaconda prompt.

Tasks in-progress

1. Creating first "Evaluator" class with basic functionality as well as the test cases for it.
2. Checking the functionality of our branches and their protection.

Calendar Week: 12 26 March, 2021

Completed Tasks

1. Added footnotes with links to the api or download pages of different libraries some of the TFE classes are based on.

2. Created package structure idea for our library in which we will create our classes.

3. Created a testing strategy based on *pytest* with three levels of testing: Unit, Integration and System.

4. Created our "code" repository in GitLab including:

    - package structure as a folders according to our system design.
    - master and develop branch with different protection settings
    - gitignore for python
    - merge request template

Tasks in-progress

1. Setting up the Visual Studio Code IDE for python and testing its functionality.

2. Setting up the anaconda environment and download the correct libraries.

Calendar Week: 11 19 March, 2021

Completed Tasks

1. Discussed with team changes on general sequence and class diagram including:
   - Changes the inheritance of the HealthIndexEstimator.
   - Changes to the general sequence diagrams with loops and aggregation of evaluations as well as configuration parsing loops.
2. Discussed the allocation and plan for the quality assurance part of our document with the team.
3. Created the UniToMultivariateWrapper class and sequence diagram.
4. Created first versions of the package structure.
5. Researched some test frameworks: unittest, DocTest and PyTest with their advantages and disadvantages.

Tasks in-progress

1. Creating the package structure and including it in the document.
2. Finishing the time series transformer and feature extractor classes and sequence diagrams with descriptions included in the document.

Calendar Week: 10 12 March, 2021

Completed Tasks

1. Included classes in tfe class diagram, also created sequence diagrams, extracted them and explained them:
   - pyts-transform-wrapper
   - pyts-impute-wrapper
   - pywt-wrapper
   - PyEMD-wrapper
2. Cleaned and aligned the general/abstract class diagram.
3. Discussed the class and sequence diagrams with the team including aligning, naming and other conventions.
4. Applied changes on the sequence diagrams: constructor part.

Tasks in-progress

1. applying changes discussed with the team on the weekly meeting including:
   - Sorting the feature/preprocessing class diagram to include timeseries to timeseries transformers.
   - Changing the general sequence diagram to include the full evaluation algorithm.

Calendar Week: 09 05 March, 2021

Completed Tasks

1. Reviewed the description of the data format.
2. Discussed changes and extension on the class/interface diagram with the team including:
   - Deciding on version B: pre defined pipelines for approaches.
   - discussing on the configuration parser and format, specifically for parameters of the approaches.
   - changes on the naming convention of interfaces and classes.
   - addition of actual classes for the different estimators and the loss/score functions.

Tasks in-progress

1. Extracting and filling feature extraction and preprocessing classes with parameters and functions.

Calendar Week: 08 26 February, 2021

Completed Tasks

1. Discussed the data format and created example ideas and explanations with team including:
    - Extending and slightly changing the Attribute Relation File Format.
    - Creating a recursive definition for time series.
    - Format design, such that all the data is represented in one file.
2. Discussed the general abstract interface diagrams with Christopher and Sanjay.

Tasks in-progress

1. describing the data format and reviewing descriptions on diagrams and format.

Calendar Week: 07 19 February, 2021

Completed Tasks

1. implemented changes according the 2nd review including:
   - addition of some more explanations for unclear parts
   - fixing general typing/grammar mistakes
   - formatting the text and images to not split text that should be together
2. added empty architecture document.
3. discussed the data format with Vinay, Saghar and Selami. We created a first draft for the data file format.

Tasks in-progress

1. writing and creating the data format and corresponding descriptions

Calendar Week: 06 12 February, 2021

Completed Tasks

1. implemented changes in TSFE chapter according to first review including:

   - creating new images without a grid background or completely new images for images taken from a paper.
   - changing textbf and underline to subsubsection and paragraph respectively.
   - fixing text mistakes denoted by the comments.

2. implemented changes in conclusion chapter according to first review including:

   - changing textbf to subsection*
   - fixing text mistakes denoted by the comments on my part of it.

3. Reviewed the whole Health State chapter.

Tasks in-progress

1. implementing 2nd review of TSFE chapter

Calendar Week: 05 05 February, 2021

Completed Tasks

1. Reviewed the RUL chapter.
2. Merged all comments from all 8 reviews of our survey into one commented pdf.
3. applied changes to the introduction chapter: structure of survey for TSFE according to review comments.
4. created and changed images in TSFE that were probably not allowed to be used; also positioning of images changed.

Tasks in-progress

1. applying changed according to comments of the review in the TSFE chapter on my parts.

Calendar Week: 04 29 January, 2021

Completed Tasks

1. Reviewed the Introduction, Conclusion and References and added comments accordingly.
2. Removed TODO from conclusion and added annotation of authors.
3. Added quotes to all chapters that were missing quotes.

Tasks in-progress

1. Reviewing of RUL chapter and adding useful comments.

Calendar Week: 03 22 January, 2021

Completed Tasks

1. discussed git structure with branches, version tags, commit message convention. (with Christopher, Vinay and Sanjay)
2. added a lot of references for methods in TSFE and images and sorted them in the .bib
3. finished the alpha version for all of my methods with explanatory images, specifically DWT, BOSS and EMD in the TSFE chapter.
4. added a short part on feature selection in evaluation setup of TSFE.

Challenges

1. Finding the right way to describe DWT was not easy.

Calendar Week: 02 15 January, 2021

Completed Tasks

1. fixed a lot of small details/mistakes in the whole chapter of the parts of TSFE.

2. created a better structure for evaluation setup in TSFE with simple example and explanation for leave-one-out cross validation.

3. discussed some ideas and structure for the conclusion chapter (with Christopher and Vinay). Also added first intro text for conclusion.

4. added first descriptions and formulas for DWT, discrete wavelet transform in TSFE.

5. added structure of the document description for TSFE chapter.

Challenges

1. Cannot the papers in DBLP easily, that I have found via Google Scholar, so I have to decide between not referencing it or using the .bib reference given by Google Scholar.

Tasks in-progress

1. Finishing alpha version of Time and Frequency Domain

2. Finding references for the methods

3. creating some pictures for explanations

Calendar Week: 01 08 January, 2021

Completed Tasks

1. Fixed the introduction and formula notation for the frequency domain in TSFE.
2. added missing bibliography references for the different methods in TSFE.
3. added description of the BOSS method in TSFE.
4. introduction for EMD method in TSFE.

Challenges

1. Finding the balance point between detail and usefulness in describing the methods.

Tasks in-progress

1. Writing and describing time and frequency domain method EMD in TSFE.

Calendar Week: 52-53 30 December, 2020

Completed Tasks

1. Discussed the new structuring and addition to our chapter TSFE (with Sanjay).

2. Extended the formal definition for TSFE specifically to also differentiate between supervised and unsupervised feature extraction.

3. Wrote the introduction and explanation of basic features for frequency domain approaches in TSFE.

Challenges

1. Adapting the notation for different equations and formulas to fit with our formal definition for Time Series and TSFE

Tasks in-progress

1. Writing and describing frequency domain feature extraction method BOSS.

Calendar Week: 51 18 December, 2020

Completed Tasks

1. Writing the basic formal definition for TSFE
2. fixing with the group the formal definition of multivariate time series.
3. extending the scope of the TSFE chapter

Challenges

1. working through and explaining the formal definition and notation of multivariate time series.

Tasks in-progress

1. extending the formal problem definition for feature extraction.
2. fleshing out the extension of the TSFE chapter.

Calendar Week: 50 11 December, 2020

Completed Tasks

1. Reading first [1] and second paper [2] on TSFE. that included:
   - explaining the python package ”tsfresh”
   - a list of 18 classification approaches for time series using distance measure algorithms
2. discussing the formal definition of time series data with the team
3. structuring our TSFE topic into bullet points and assigning and annotation authors to those sections (with Sanjay)
4. creating a weekly plan for writing the chapter on TSFE. (with Sanjay)

Challenges

1. understanding how or what feature extraction method to extract from the distance measure algorithms

Tasks in-progress

1. writing the formal problem definition for feature extraction.

References

[1] Christ, Maximilian, et al., ”Time series feature extraction on basis of scal-able hypothesis tests (tsfresh–a python package)”. *Neurocomputing*, 307(2018): 72-77., https://www.sciencedirect.com/science/article/pii/S0925231218304843

[2] Bagnall, A., et al., The great time series classification bake off: An ex-perimental evaluation of recently proposed algorithms. Extended version.arXiv 2016.”. *arXiv preprint arXiv:1602.01711.*, https://arxiv.org/pdf/1602.01711.pdf

Calendar Week: 49 4 December, 2020

Completed Tasks

1. Planning how we work on the topic ”Times Series Feature Extraction” (TSFE) (with Sanjay).

2. Created ”topic study” folder and added template for our group-survey in our documentation repository [3].

3. Working with the team on decisions for the template, specifically how to annotate the author to the sections and subsections.

Challenges

1. Understanding how we have to adapt the chapter outline for the topic TSFE, as this topic differs somewhat from the other three topics.

Tasks in-progress

1. Reading first [1] and second paper [2] listed on the ”Milestone 2: Topics” document for our topic TSFE.

References

[1] Christ, Maximilian, et al., ”Time series feature extraction on basis of scal-able hypothesis tests (tsfresh–a python package)”. *Neurocomputing*, 307(2018): 72-77., https://www.sciencedirect.com/science/article/pii/S0925231218304843

[2] Bagnall, A., et al., The great time series classification bake off: An ex-perimental evaluation of recently proposed algorithms. Extended version.arXiv 2016.”. *arXiv preprint arXiv:1602.01711.*, https://arxiv.org/pdf/1602.01711.pdf

[3] https://git.cs.uni-paderborn.de/machine-learning-for-predictive-maintenance/documentation.

Calendar Week: 48 27 November, 2020

Completed Tasks

1. Finished reading third survey [1], which:
   - summarizes the application of RF, ANN, SVM (also SVR) and K-means in the field of PdM.
   - proposes and uses a literature review planning protocol for the papers on PdM.
   - meta-analyses these papers (e.g. citation and methods analysis, distribution along the years).

Tasks in-progress

1. Reading dissertation [2].

References

[1] T. P. Carvalho and F. Soares and R. Vita and R. Francisco and João P. Basto and Symone G.S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance”, *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[2] J. K. Kimotho, “Development and performance evaluation of prognostic ap-proaches for technical systems,”. Ph.D. dissertation, 2016, https://digital.ub.uni-paderborn.de/hsx/content/titleinfo/2219021.

Calendar Week: 47 20 November, 2020

Completed Tasks

1. Finished reading second survey [2], which:
    - explains the general framework and workflow of a PdM-system using the example of washing equipment.
    - explained three classic ML models: LR, SVM and DT/RF shortly and creates a big overview of their application in PdM in the literature.
    - explaines three neural network method: ANN, DNN and AE and their application in Pdm similar to above.

Tasks in-progress

1. Reading third survey [1].

References

[1] T. P. Carvalho and F. Soares and R. Vita and R. Francisco and João P. Basto and Symone G.S. Alcalá, “A systematic literature review of machine learning methods applied to predictive maintenance”, *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[2] W. Zhang and D. Yang and H. Wang, ”Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey”, *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213-2227, 2019, https://ieeexplore.ieee.org/document/8707108.

Calendar Week: 46 13 November, 2020

Completed Tasks

1. Fixed my weekly report according to our discussion (references per week instead of for the whole document) and removed unnecessary .bib file from our documentation repository [3].

2. Finished reading first survey [1], which includes:

   - a high level view of Maintenance through broad categories: PdM, PM and RM.
   - an overview of system architectures for PdM like Maintenance 4.0, Cloud-based approaches and OSA-CBM.
   - a presentation of different objectives of PdM: cost minimization, availability and reliability maximization and multi-objective optimization.
   - a review and extensive exploration of the application of knowledge based, classical ML based and Deep Learning based methods aswell as hybrid approaches in the field of PdM.
   - a short list of future research areas for PdM.

Challenges

1. Not getting overwhelmed by the number of different approaches and applications of ML methods in the topic of PdM.

Tasks in-progress

1. Reading second survey [2].

References

[1] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, ”A survey of predictive maintenance: Systems, purposes and approaches”, *arXiv preprint arXiv:1912.07383*, 2019, https://arxiv.org/pdf/1912.07383.pdf.

[2] W. Zhang and D. Yang and H. Wang, ”Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey”, *IEEE Systems Journal*, vol. 13, no. 3, pp. 2213-2227, 2019, https://ieeexplore.ieee.org/document/8707108.

[3] https://git.cs.uni-paderborn.de/machine-learning-for-predictive-maintenance/documentation.

Calendar Week: 45 06 November, 2020

Completed Tasks

1. Created Gitlab structure for the documentation-project.
   started with simple folder structure for minutes and weekly reports:
   for both, a folder with a subfolder for the templates.

2. Added a subgroup for the milestones on GitLab.

3. Connected GitLab Updates to the Microsoft Teams Group (with Christopher).

Challenges

1. Connecting GitLab to Sourcetree on my computer was more difficult than expected.

Tasks in-progress

1. Reading first survey [1].
   wrote down information about the general topics of predictive maintenance: system architectures, objectives and subcategories of fault diagnosis & prognosis.

References

[1] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, ”A survey of predictive maintenance: Systems, purposes and approaches”, *arXiv preprint arXiv:1912.07383*, 2019, https://arxiv.org/pdf/1912.07383.pdf.