Plan Overview

A Data Management Plan created using DeiC DMP

Title: Ph.D project - Python-based EASETECH waste and waste prevention modelling

Creator: Emil Laurberg Pedersen

Project Administrator: Anders Damgaard, Marie Kampmann Eriksen

Affiliation: Danmarks Tekniske Universitet / Technical University of Denmark

Template: DTU DMP Template

Project abstract:

The project will focus on transferring EASETECH into a Python-based environment to counter current waste modeling limitations, e.g. modeling temporal and geographical dimensions when assessing the lifecycle of waste management systems. Furthermore, the project focuses on expanding the scope of waste systems to include the influence of design and use on waste, before being generated at End-of-Life, allowing for modeling waste prevention strategies, consumer influence and policy changes.

ID: 8254

Start date: 01-06-2026

End date: 31-05-2029

Last modified: 17-08-2026

Ph.D project - Python-based EASETECH waste and waste prevention modelling

Planning checklist

Policies, guidelines, procedures and best practices that are followed

- DTU Research Data Management Policy
- DTU Danish Code of Conduct for Research Integrity
- DTU Information Security Policy
- Local Guidelines and Procedures
- Agreement with project partners
- Funder requirements

Other

No additional Policies, guidelines, procedures and best practices.

Data collection

Information about the datasets you will create in the project

Dataset (name, ID)	Data type and format	Data collection method/data origin	DTU Information Security Classification	Equipment/software/instruments involved in data collection (e.g. instrument ID for instrument listed in EIS)	File naming convention and versioning	Folder structure	Expected/estimated volume (MB, GB, TB)	Objective
Data_model_1: Repair	Algorithm/method, digital data - formatted as python code and structured excel data	Data will be developed as python code.	Level 1 until published, then Level 0	Python version 3.10	Versioning is done in the following structure: Model Version X.Y.Z. Major updates, influencing modeling workflow, method and results updates the X. Minor updates not affecting modelling and results, but adds features and updated functions to the model updates the Y. Small updates like updating data definitions and descriptions and configuration/installation files will update the Z. Description of updates are added in GitHub.	folder structure will include folders for data, results, scripts, templates and/or tutorials. Furthermore, in the root directory, there will be a README and 2 versions of how to setup the model virtual environment. User generated data is stored within the cloned, local root directory, and will not be uploaded, collected or monitored in any way by others.	10 MB	Present a framework as the product of research.
Case_study_data_1: Repair	Structured digital excel data	Data will be collected using desktop research or will be obtained from project partners.	Level 1	Excel	Versioning is done in the following structure within the excel README sheet: Data Version X.Y Major updates, influencing data structure, method and adding new data updates the X. Minor updates like updating data definitions and descriptions and README sheet will update the Y. In the README file, a version history and changes description will be present and continuously updated.	Structured excel sheets with labels for raw and processed data. Every structured excel sheet contains a structured README sheet, explaining the data usage, origin and handling	Approx. 1.6 GB	Case study, supporting and demonstrating the framework
Data_model_2: Py-EASETECH	Algorithm/method, digital data - formatted as python code and structured excel data	Data will be developed as python code.	Level 1 until published, then Level 0	Python version 3.10, Brightway2.5, EASETECH version 3.6.11	Versioning is done in the following structure: Model Version X.Y.Z. Major updates, influencing modeling workflow, method and results updates the X. Minor updates not affecting modelling and results, but adds features and updated functions to the model updates the Y. Small updates like updating data definitions and descriptions and configuration/installation files will update the Z. Description of updates are added in GitHub.	Folder structure will include folders for data, scripts and tutorials. Furthermore, in the root directory, there will be a README and 2 versions of how to setup the model virtual environment. Lastly, user-generated data will not be stored in this folder,but in an external, self chosen folder.	50 MB	Present a framework as the product of research.

Case_study_data_2: Py-EASETECH	Structured digital excel data	Data will be collected using desktop research or will be obtained from project partners.	Level 1	Excel	Versioning is done in the following structure within the excel README sheet: Data Version X.Y Major updates, influencing data structure, method and adding new data updates the X. Minor updates like updating data definitions and descriptions and README sheet will update the Y. In the README file, a version history and changes description will be present and continuously updated.	Structured excel sheets with labels for raw and processed data. Every structured excel sheet contains a structured README sheet, explaining the data usage, origin and handling	500 MB	Case study, supporting and demonstrating the framework
Data_model_3: Waste_prevention	Algorithm/method, digital data - formatted as python code and structured excel data	Data will be developed as python code.	Level 1 until published, then Level 0	Python version 3.10, Brightway2.5, EASETECH version 3.6.11	Versioning is done in the following structure: Model Version X.Y.Z. Major updates, influencing modeling workflow, method and results updates the X. Minor updates not affecting modelling and results, but adds features and updated functions to the model updates the Y. Small updates like updating data definitions and descriptions and configuration/installation files will update the Z. Description of updates are added in GitHub..	Folder structure will include folders for data, results, scripts, templates and/or tutorials. Furthermore, in the root directory, there will be a README and 2 versions of how to setup the model virtual environment	25 MB	Present a framework as the product of research.
Case_study_data_3: Waste_prevention	Structured digital excel data	Data will be collected using desktop research or will be obtained from project partners.	Level 1	Excel	Versioning is done in the following structure within the excel README sheet: Data Version X.Y Major updates, influencing data structure, method and adding new data updates the X. Minor updates like updating data definitions and descriptions and README sheet will update the Y. In the README file, a version history and changes description will be present and continuously updated.	Structured excel sheets with labels for raw and processed data. Every structured excel sheet contains a structured README sheet, explaining the data usage, origin and handling	500 MB	Case study, supporting and demonstrating the framework

Will you reuse any existing data (including personal data)?

- Yes.

Existing open-source background data from EASETECH will be reused (link: Forside - Easetech). Additional external datasets may be reused depending on case-study selection.

Legal aspects

Use and reuse of all or parts the data are limited by:

At this point in time, every piece of data is expected to be open source and therefore not limited by any type of agreement.

Describe in the text field any agreements with collaborators and restrictions that can limit the use, reuse, sharing and publication of data.

Currently no agreements that limit the use, reuse, sharing and publication of data.

Ethics checklist

The project does not require any ethical review.

Will you use personal data in the project?

- No

As of now, no personal data will be used.

Will you use special categories of personal data?

- No

As of now, no personal data will be used.

Is personal data collected by you or by others?

Are you receiving personal data from others?

- No

Are you transferring personal data to others?

- No

No personal data collection is part of this project.

Data storage

Description of storage solutions.

Dataset and datatype	Data storage service	Back – up procedure (to ensure no data is lost)	Folder and file path
Data_model_1: Repair	GitHub	GitHub version storing and offline backups on DTU cloud drive	Secure github repository, cloned down using GitHub software to an example data repository: "C:\Users\elape\OneDrive - Danmarks Tekniske Universitet\Dokumenter\GitHub\SMC-R"
Case_study_data_1: Repair	DTU cloud storage	Cloud backup and M drive for local backups	SharePoint Address: "https://dtudk.sharepoint.com/:f:/r/sites/ELAPE_PHD/Delte%20dokumenter/General/PhD_case_studies/Case_study_1" M drive address: "M:/PhD/Articles/Article1_SMC-R/Case_study_1_data"
Data_model_2: Py-EASETECH	GitHub	GitHub version storing and offline backups on DTU cloud drive	Secure github repository, cloned down using GitHub software to an example data repository: "C:\Users\elape\OneDrive - Danmarks Tekniske Universitet\Dokumenter\GitHub\Py-ET"
Case_study_data_2: Py-EASETECH	DTU cloud storage	Cloud backup and M drive for local backups	SharePoint Address: "https://dtudk.sharepoint.com/:f:/r/sites/ELAPE_PHD/Delte%20dokumenter/General/PhD_case_studies/Case_study_2" M drive address: "M:/PhD/Articles/Article2_Py-ET/Case_study_2_data"
Data_model_3: Waste_prevention	GitHub	GitHub version storing and offline backups on DTU cloud drive	Secure github repository, cloned down using GitHub software to an example data repository: "C:\Users\elape\OneDrive - Danmarks Tekniske Universitet\Dokumenter\GitHub\WastePrev"
Case_study_data_3: Waste_prevention	Sharepoint	Cloud backup and M drive for local backups	SharePoint Address: "https://dtudk.sharepoint.com/:f:/r/sites/ELAPE_PHD/Delte%20dokumenter/General/PhD_case_studies/Case_study_3" M drive address: "M:/PhD/Articles/Article3_wasteprevention/Case_study_3_data"

Description of the solutions used for data sharing with colleagues and collaborators

Collaborators: Sharing data will be done using SharePoint using tracked changes and version storing. For the model data, sharing data will be done by adding the collaborators as contributors for the GitHub repository.
For colleagues and reviewers, access will be granted according to project role and need-to-know principles and removed when the collaboration ends. This will still be done using SharePoint and GitHub.

Procedure for granting access to collaborative data in a secure way

SharePoint invitation to a limited, chosen group of collaboratives after mutual agreement and clear definition of roles.

Documentation

Indicate the documentation/information necessary to read, interpret and/or reproduce the data:

- Jupyter Notebooks will be used to provide a step-by-step tutorial and description for the user to understand the data, method and results.
- ReadMe files are used to provide data source, context, licenses and guidance on navigating and using the data, along with its scope, features and limitations.
- Articles are used to present the theoretical, academic and technical background and reasoning for why the data and methods have been developed.

The type of presented documentation below will be carried out for each relevant data model and case study to ensure proper documentation. The table will be filled out gradually once the models and files have been developed.

Type of documentation	**Title**	**Storage location**	**Version control**	**Format**
Jupyter Notebook	TBD	Within the parent folders of each developed data model.	GitHub	.ipynb
ReadMe file	README	Within the parent folders of each developed data model.	GitHub	.md or .txt
Excel ReadMe sheets	TBD (sheet name: README)	Within the excel sheet in question	Manual version control within the README sheet	.xlsx
Articles	TBD	DTU FindIt and other publication pages	-	.pdf

Metadata

Which metadata will be provided to enrich the data?

README files will follow a consistent project-specific structure including dataset description, provenance, variable definitions, units, assumptions, software dependencies and version history. Semantic Versioning will be used for software releases. GitHub repositories will include CITATION.cff files to support easy citation.

Dataset	**List data and metadata standards that you will use** (e.g. DataCite metadata, dicsipline specific controlled vocabulary etc.)	**Will metadata be generated automatically or entered manually?**
Data_model_1: Repair	DataCite metadata schema for published repository releases (DOI), CITATION.cff, GitHub repository metadata, README documentation, semantic versioning	Automatic and manual
Case_study_data_1: Repair	DataCite metadata schema, Excel README sheet, controlled variable names, units and definitions documented in README	Manual
Data_model_2: PyEASETECH	DataCite metadata schema, CITATION.cff, GitHub metadata, README documentation, semantic versioning	Automatic and manual
Case_study_data_2: PyEASETECH	DataCite metadata schema, Excel README sheet, controlled variable and unit definitions	Manual
Data_model_3: Waste Prevention	DataCite metadata schema, CITATION.cff, GitHub metadata, README documentation, semantic versioning	Automatic and manual
Case_study_data_3: Waste Prevention	DataCite metadata schema, Excel README sheet, controlled variable and unit definitions	Manual

Data publication

List of datasets/outputs of the project, how they will be published and follow the FAIR principles.

Dataset (title/name)	**Will data be published open access** (yes/no)	**When will data be openly available**	**Embargo period/access level** (specify if data will be open access after an embargo period)	**Are the dataset supporting data for a publication** (yes/no)	**In which repository do you intend to publish data**	**Accompanying documentation, tools or software needed to create, process or visualize the data**	**How will potential users find the data**	**DOI** (Insert DOIs for published datasets)
Data_model_1: Repair	yes	Available together with article (expected: 3/1/2027-6/1/2027)	No embargo period	yes	GitHub	README, installation guide, tutorials, CITATION.cff, Python environment specification (requirements.txt/environment.yml)	Linked from publication, GitHub repository URL, repository metadata, CITATION.cff file	DOI generated automatically through GitHub-Zenodo integration when released.
Case_study_data_1: Repair	no	Restricted access when article is published	Metadata openly available but access may be restricted for third-party datasets	yes	DTU Data where permitted	README, variable descriptions, units, source references	DOI and metadata linked from publication	To be assigned if published
Data_model_2: PyEASETECH	yes	Available together with article (expected: 3/1/2028-5/1/2028)	No embargo period	yes	GitHub	README, installation guide, tutorials, CITATION.cff, Python environment specification (requirements.txt/environment.yml)	Linked from publication, GitHub repository URL, repository metadata, CITATION.cff file	DOI generated automatically through GitHub-Zenodo integration when released.
Case_study_data_2: PyEASETECH	no	Restricted access when article is published	Metadata openly available but access may be restricted for third-party datasets	yes	DTU Data where permitted	README, variable descriptions, units, source references	DOI and metadata linked from publication	To be assigned if published
Data_model_3: Waste Prevention	yes	Available together with article (expected: 1/1/2029-4/1/2029)	No embargo period	yes	GitHub	README, installation guide, tutorials, CITATION.cff, Python environment specification (requirements.txt/environment.yml)	Linked from publication, GitHub repository URL, repository metadata, CITATION.cff file	DOI generated automatically through GitHub-Zenodo integration when released.
Case_study_data_3: Waste Prevention	no	Restricted access when article is published	Metadata openly available but access may be restricted for third-party datasets	yes	DTU Data where permitted	README, variable descriptions, units, source references	DOI and metadata linked from publication	To be assigned if published

Are there any legal or ethical protection that limits the access of data or subsets of data? (e.g. IPR, Copyright, dual use, GDPR)

- No

Currently, no limited access to data or subsets.

Are the datasets connected to other research outputs and/or resources ? (e.g. research papers, reports, posters, instruments, High performance computing)

Yes - datasets are connected to PhD articles, and governmental reports (published EU and Danish EPA reports). Connected to GitHub repositories and connected Python packages.

Costs and resources

What resources will be dedicated to data management?

- Hours

Costs regarding upkeep and ensuring data is usable and applicable when other model parts are updated. This work is not constant and requires variable time to do.
Data storage is managed and stored by DTU IT and no costs is assumed in this connection.

Long-term preservation

Specify how data will be preserved beyond the scope of the projects and how the value will be secured:

What is the total amount of data collected (in MB/GB/TB)?	~500 MB
What procedure will be used to handle requests for data? (in case data access are restricted)	Requests for restricted data should be followed up with a written NDA and accept that the published data and results cannot be retraced or reverse engineered to the restricted dataset.
Which data will be preserved? (specify criteria for long term preservation (>5-10 years))	All project-generated code, documentation and publishable datasets will be preserved. Externally provided datasets will be preserved according to applicable agreements and licensing conditions.
Where is data archived?	GitHub, Zenodo and DTU managed storage, saved on M drive and managed by DTU IT.
Which documentation created during the project is necessary to interpret the data? (NB: Some types of State research data must be reported to The Danish National Archives).	README files, Jupyter notebooks, tutorials and associated publications.
Where is documentation stored?	Together with data: GitHub and DTU repositories, saved on M drive and managed by DTU IT.
How will readability of the data be guaranteed?	With README files, tutorials and repository explanation.
Which data will be deleted and when?	No data is expected to be deleted. Data will only be deleted if seen as wrong or not usable whatsoever.

Data contact persons

Specify names, roles and responsibility for data in the project

People involved in data management (name)	Specify role (e.g. data collector, data analyst)	Contact info (mail and/or ORCID)	Who is responsible for archiving or deleting data?	Where is the final DMP stored and who has access to it?
Anders Damgaard	Supervisor	adam@dtu.dk	Yes	The DMP is public and everyone can access it. Only PhD student, co-supervisor and supervisor can edit
Emil Laurberg Pedersen	PhD student	elape@dtu.dk	Yes	The DMP is public and everyone can access it. Only PhD student, co-supervisor and supervisor can edit

Change history

Indicate version and changes to the DMP

Version and date	Changed by	Changes
V1.0.0	Emil Laurberg Pedersen	Created DMP