Bridging the Modeling Gap: Automating Data Flow from PyPSA to EnergyScope Semester Project

presented by
Till Cassens
StudentID no. 24-942-880

Supervisor:
Gabriel Anselm Wiest, M.Sc.

Examiner: Prof. Dr.-Ing. André Bardow

Zürich, December 16, 2025

Abstract

read the paper "A guide to writing articles in energy science" from Martin Weiss & Alexandra M. Newman (2011)

Abstract and Introduction The abstract (short summary) is placed at the very beginning of the work (after the table of contents). It comprises at most one page. In the abstract, only the following questions are of interest: What is your work about (question)? Which methods were used (solution)? What is the outcome (results)? A reader who has read the abstract should know the rough outlines of the work and its most important results (no details!). Not included in the abstract are the motivation, history of the problem, or information about the structure of the work Usually, the abstract does not contain equations, figures, tables, literature citations, or other references to other pages of the paper The introduction marks the beginning of the actual work. It leads to the research question by showing the bigger picture of the research and placing it in relation to existing knowledge (literature). In contrast to the abstract, the introduction, therefore, also includes statements on the motivation and background of the work. The introduction then defines the scientific questions to be examined in the context of the work and mentions the methods used or further developed to solve the problem. The last section of the introduction gives an overview of the structure of the work ("storyline"). A brief description suffices of the steps of the research described in each chapter and how the chapters build on each other. However, no results are presented yet. Only the procedure is explained!

Contents

List of Figures

List of Tables

1. Introduction

The ongoing transition from a centralized fossil fuel dominated energy system to a decentralized and renewable based one represents a fundamental change in how energy infrastructures are planned and operated. Although declining costs of solar and wind power support this shift, their stochastic and weather dependent behaviour differs significantly from the controllable and dispatchable fossil based generation that has historically ensured system stability. This increasing variability, together with a growing integration of electricity, heating and mobility sectors, has made the design and operation of modern energy systems considerably more complex. To address these challenges, energy system optimization models (ESOMs) have become essential tools for analysing technological options, evaluating policy measures and supporting long term investment decisions. [1]

In response to this need, a broad landscape of modeling frameworks has emerged and is routinely applied in academia, industry and the public sector. However, as several recent reviews highlight, the growing detail and scope of modern models also introduce significant challenges. [2, 3] Among the most important are high computational requirements, the need for transparent and reproducible workflows, and the considerable effort required for collecting and preprocessing input data. In many cases, this data related effort becomes a primary barrier to broader model adoption.

A model specifically designed for fast scenario analysis while remaining easy to use is the EnergyScopeTD framework. [4] Originally released as an online calculator to enhance the energy literacy of Swiss citizens by allowing them to explore energy system interactions, it has since evolved into the linear programming optimization framework used in research today. According to its developers, EnergyScope TD was designed with accessibility as a core objective. As stated by Limpens et al. (2019), the model "can be run out of the box" and provides "free and open source code with a simple formulation to make it accessible to all". Running a model truly out of the box requires all input data to be preprocessed and openly available. For EnergyScopeTD, this is currently the case only for Switzerland, Italy and Belgium. For these countries, national models were created through manual data collection from various sources and technology databases. Extending this process to additional countries requires extensive data gathering and harmonisation, which limits the model's broader deployment.

To overcome this limitation and to enable more widespread use of EnergyScopeTD, this semester project develops an open source workflow that automatically generates all required input data for all European countries. The goal is to allow researchers and practitioners to use EnergyScope TD for scenario analysis across Europe without engaging in manual data collection.

The workflow developed in this project is not constructed entirely from the beginning but instead builds on an existing and widely adopted open source framework. Interviews with modeling experts and comparative studies identify Python for Power System Analysis (PyPSA) [5] as one of the leading tools offering a transparent data processing pipeline. [6] In particular, the PyPSA-Eur model [7] provides a comprehensive and well documented workflow for preparing energy system data for the European transmission grid. Many of the processing steps required for EnergyScopeTD, such as collecting demand data, processing land restrictions or preparing time series, already exist within PyPSA-Eur, making it a suitable foundation for a higher level workflow. Building on this foundation, the workflow developed in this project adds a dedicated layer that translates the processed data into the specific structure and format required by EnergyScopeTD. This ensures consistency, avoids duplication of effort and benefits from the actively maintained PyPSA-Eur ecosystem. At the same time, it enhances the usefulness of EnergyScopeTD, which complements high resolution frameworks such as PyPSA by enabling fast sensitivity analyses, uncertainty studies and the exploration of many scenarios that would be computationally demanding with more detailed models.

In the following chapter, the setup and differences between the two models are described, along with the data workflow implemented in PyPSA-Eur. Chapter 3 introduces the methodology and implementation of the workflow developed for this semester project, followed by a discussion of the results in Chapter 4. Finally, Chapter 5 concludes the thesis and highlights areas for future development.

2. System and Data Overview

This chapter first introduces the concept of energy system optimization models in a broad sense and then focuses on the input data required for such models. The remainder of the chapter describes the two models used in this semester project and provides a detailed overview of their data workflows, which serve as the foundation for the subsequent chapters.

2.1. Energy System Optimization Models (ESOMs)

Energy system optimization models (ESOMs) are used today across a wide range of applications, including the optimal scheduling of power plants, the assessment of policy measures, and the planning of infrastructure expansion. Their general purpose is to analyse the balance between energy supply and demand. [8] ESOMs commonly follow a bottom-up modelling philosophy, where technologies are represented with detailed technical and economic parameters. In their simplest implementation, they rely on linear programming to optimise the system according to a chosen objective. These models are additionally subject to various constraints, arising either from physical laws or from expert judgments by the modeller, in order to produce realistic results.

Mathematically, such models can be represented in the following way:

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & f(x) \\ \text{subject to} \quad & h(x) = 0 \\ & g(x) \leq 0 \end{aligned}$$

where $f(x)$ is the objective function (for example, minimizing system costs) and $h(x)$ and $g(x)$ represent equality and inequality constraints that must be satisfied (such as electricity demand or specific technological behaviour). Depending on the formulation of the objective function, the constraints and the decision variables, the optimization problem can be linear (LP), mixed-integer linear (MILP), nonlinear (NLP) or mixed-integer nonlinear (MINLP). [9] A detailed discussion of these models and their mathematical formulation can be found in [10].

The focus of this project lies on the input data required to formulate realistic model constraints and thereby enable meaningful representations and analyses of the energy system. Because such constraints rely directly on the underlying assumptions and data sources, the availability and transparency of input data play a crucial role in shaping model outcomes.

2.2. Open Data for ESOMs

ESOMs are inherently assumption based, and their outcomes can differ significantly depending on the assumptions chosen by the developer. [11] Beyond explicit modelling choices, such as how a specific technology is represented and at what level of detail, the input data itself often embeds implicit assumptions that strongly influence model results.

ESOMs require several types of data to run. This data can typically be divided into three categories: time series data (for example renewable generation profiles), geographic data (such as power plant locations or maximum installable renewable capacities) and tabular data (including technology costs). [12] These datasets are usually collected from multiple sources, requiring extensive harmonisation and preprocessing before they can be used in a model. Although some of the data is publicly available, the collection process is often time-intensive and prone to error. Assumptions made during preprocessing are frequently undocumented, making the resulting workflow difficult or impossible to reproduce. [13]

For these reasons, many authors have argued for years that open data and open-source modelling workflows are not merely "nice to have", but essential for ESOMs. Given the profound societal implications of these models, especially in the context of energy and climate policy and the inherent difficulty of verifying model outcomes, full transparency of both model formulation and input data is necessary. [14] These efforts converge in the Open Energy Modelling (openmod) initiative[1], a community of leading researchers advocating for fully open energy models and openly accessible data.

[1] See https://openmod-initiative.org for details.

Figure 2.1 illustrates this idea by distinguishing between open data, open source and open access. Since this project focuses on the preparation of input data, it primarily relates to the first part of the workflow.

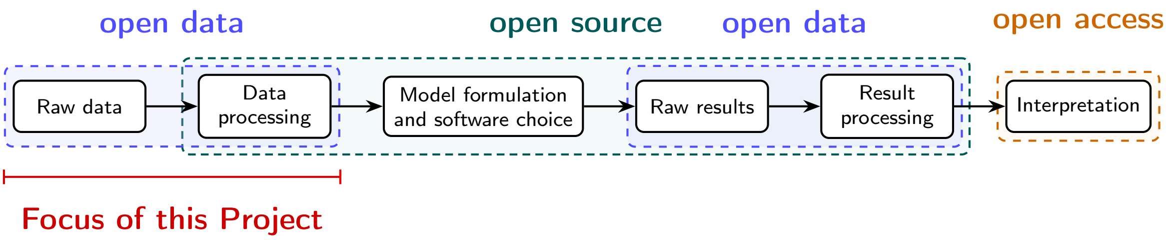

Figure 2.1.: Open data, open source and open access ESOMs, adapted from [12].

Beyond meeting scientific standards such as reproducibility, transparency and peer review, open data and open source code also offer practical benefits. As Pfenninger et al. [11] argue, open practices can increase visibility and readership for the researcher, while providing broader advantages for the research community. Shared datasets reduce duplication of effort, allow others to build on existing work and ultimately increase productivity. In this context, the term "open" also encompasses the associated licences that ensure data and code can be freely reused, such as MIT or other permissive licences.

2.3. EnergyScopeTD

EnergyScopeTD is an LP optimization framework that models and optimizes the entire energy system for a target future year with an hourly resolution, considering both investment and operation. The model is mostly used in a greenfield approach. In addition to the LP formulation, it incorporates a typical-day selection method, which drastically reduces the computational time of the optimization problem. [4]

As outlined previously, EnergyScopeTD distinguishes itself from other energy system optimization frameworks by covering all major energy sectors while being explicitly designed for fast scenario analysis. The core of the model is implemented in AMPL,[2] a high-level programming language tailored to reflect the structure and simplicity of mathematical formulations.

[2]See https://ampl.com/ for details.

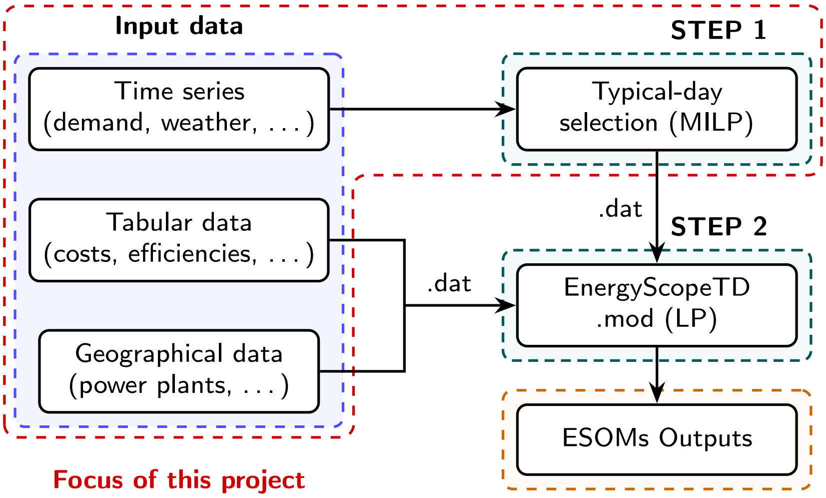

Figure 2.2.: Two-step EnergyScopeTD workflow, adapted from [4].

Figure 2.2 illustrates the two-step workflow of EnergyScopeTD and highlights the components that are integrated into the workflow developed in this project. The framework relies on three essential files: the model file (.mod), which contains the full set of equations and constraints, the typical-day time-series file (.dat) and a data file (.dat) that provides all remaining model parameters required for the complete formulation of the problem. The logical relationships within the workflow, together with the corresponding file types (.dat or .mod), are also indicated in the figure.

The EnergyScopeTD model itself is organised into three main parts: resources, energy conversion and demand. This structure is illustrated for a simplified energy system in Figure 2.3. The end-use demands that the model must satisfy include heat, electricity

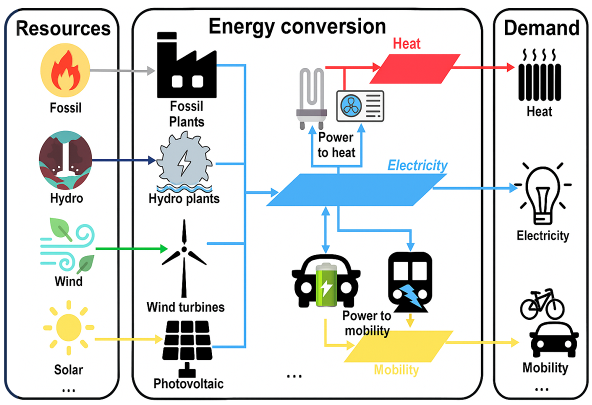

Figure 2.3.: Simplified Structure of EnergyScopeTD, taken from [4]

and mobility. In more detailed applications, additional non-energy demands or a finer disaggregation of end-use categories can also be represented. For each end-use type the model defines a dedicated layer that balances all incoming energy flows on the supply side and all outgoing flows on the demand side. Conversion technologies connect these layers by transforming one carrier into another and thereby enable sector coupling within the system. The model also includes storage technologies that withdraw energy from a layer and return it at a later time step, which provides temporal flexibility. In addition to the end-use layers, EnergyScopeTD represents intermediate energy carriers that are not final energy services, such as wood or other biomass resources, in order to model upstream resource flows in a consistent manner. [4]

2.4. PyPSA-Eur

The core optimisation framework of Python for Power System Analysis (PyPSA) was developed to harmonise steady-state power-flow tools with energy system optimisation models (ESOMs), thereby enabling more systematic investigations of future grid expansion needs [5]. Building on this foundation, several regional models have been created for different parts of the world. The model relevant for this project is PyPSA-Eur, which represents the high-voltage transmission grid of Europe [15, 7]. Figure 2.4 illustrates the base network underlying PyPSA-Eur.

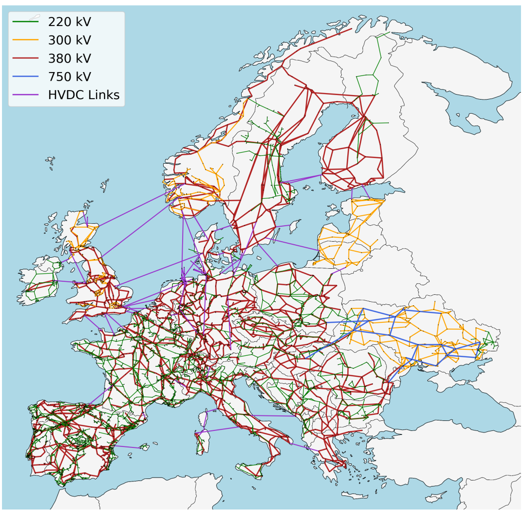

Figure 2.4.: Base network of the European transmission grid represented in PyPSA-Eur [7].

Each line in the network represents a transmission line connecting nodes, which correspond to real-world substations in the European grid. These substations interface with the underlying distribution networks, although the distribution level is not explicitly modeled in PyPSA. PyPSA-Eur was initially developed as an electricity-only model, but it has gradually been extended to include additional energy carriers and sectors. As a result, it now provides a representation of a fully sector-coupled, multi-energy system. [15]

All demands and system attributes are assigned to the local nodes, typically after spatial clustering to reduce the number of nodes and thereby the computational burden of large-scale optimisation. [5] The details of this preprocessing workflow are presented in the next section.

PyPSA describes the energy system using a modular component structure, shown in Figure 2.5.

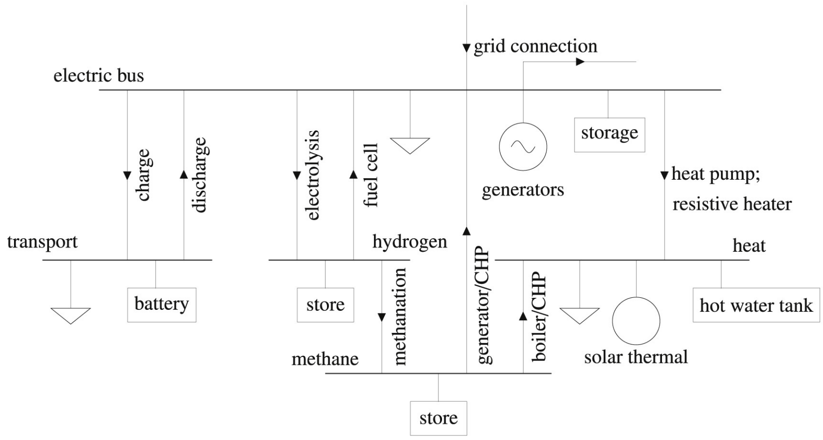

Figure 2.5.: Simplified structure of a single node in PyPSA, taken from [7].

Each node contains multiple buses that balance energy carriers such as electricity, hydrogen, methane and heat. Generators, loads and storage technologies are connected to these buses and supply, consume or buffer energy over time. Conversion technologies such as electrolysers, fuel cells, boilers, CHP units and heat pumps are modelled as links between buses, enabling the transformation of one energy carrier into another and thereby facilitating sector coupling across the system.

This modular and generic architecture is flexible enough to represent the complexity of modern multi-energy systems, while also allowing users to introduce additional technologies or adapt existing ones. [15]

2.5. Data Workflow PyPSA-Eur

All input data required to formulate the optimization model and represent an energy system in PyPSA-Eur is derived from open-source datasets. These sources are preprocessed and harmonized into files in the NetCDF (Network Common Data Form) format. The final NetCDF file describes the entire network, including all constraints and potentials and can be used by the PyPSA framework to solve the optimization problem.

The preprocessing workflow can be divided into three main steps: (1) construction of the base network, (2) simplification and clustering of the base network and (3) mapping of additional data to the resulting clusters. Figure 2.6 illustrates this workflow, including example outputs for each step.

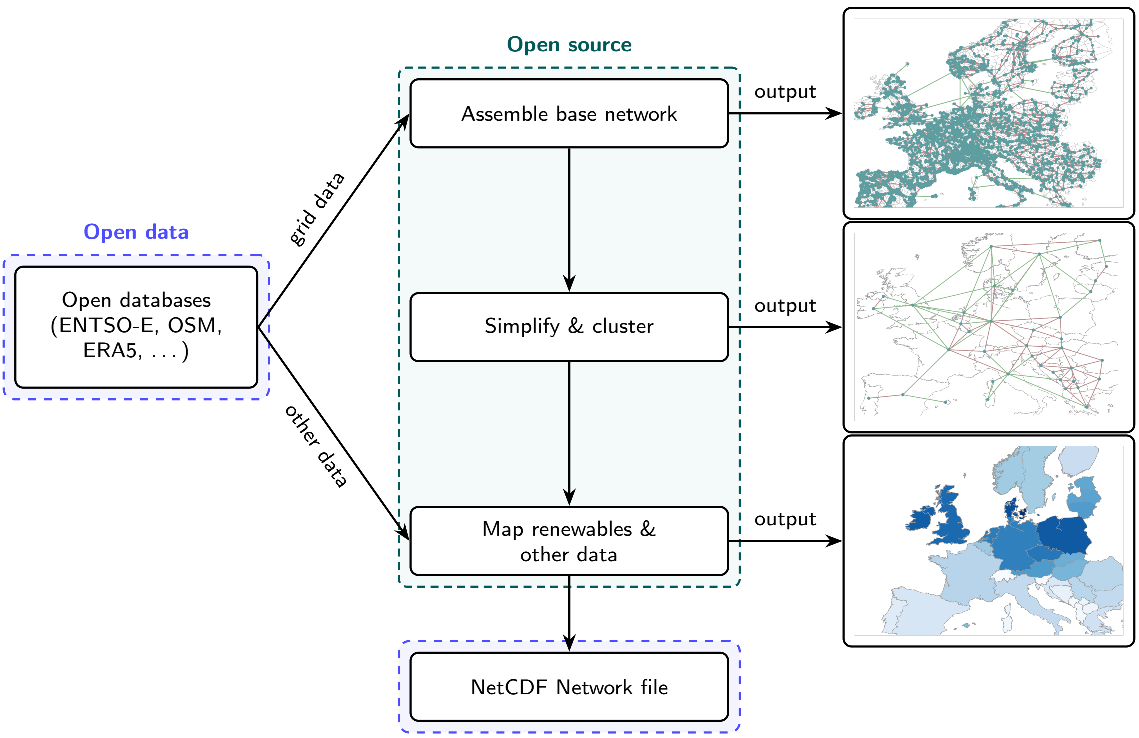

Figure 2.6.: Simplified PyPSA-Eur Data Processing Workflow

Almost all open data used in the model is georeferenced and will be assigned to the nearest node within the same country. To enable this, a Voronoi cell is constructed around each clustered node, encompassing all points that are closest to that specific node. This provides an efficient method for assigning all relevant features to their appropriate locations.

The entire workflow consists of numerous sub-steps that perform tasks such as data collection, harmonization and the mapping of attributes to specific nodes. The sequence and execution of these steps are fully automated by the Snakemake workflow management system [16]. Each step in the workflow is represented by a rule, which specifies how to compute a given output file from designated input files using a predefined script. An simple example of such a rule is shown in Figure 2.7.

```
rule plot_results:
    input:
        "input/data.csv"
    output:
        "plots/results.svg"
    script:
        "scripts/plot.py"
```

Figure 2.7.: Example Snakemake rule for plotting results

All rules are automatically linked and executed in the correct order, such that the output file of one rule becomes the input file for the next. Snakemake enables the construction of human readable workflows that are reproducible, scalable and easy to maintain. [16] PyPSA Eur provides a wide range of configuration options for the workflow, for example specifying which land types should be excluded for certain technologies. For further details, please refer to the official documentation.[3]

In the remainder of this section, several key data processing steps are described in detail. These steps are later used in the workflow developed in Chapter 3.

2.5.1. Renewable Technologies Time Series

In the configuration file for the workflow, a specific weather year can be selected, which is then used to calculate the capacity factor time series for renewable energy technologies. For the selected year, the corresponding weather data is processed and converted into time series using the atlite package [17]. Multiple sources can be used as raw weather data, with the most common and fully integrated options being ERA5 [18] and SARAH 2/3 [19].

Wind For the wind power capacity factors a specific reference turbine must be selected (the default corresponds to a turbine with a nominal power of 3 MW). Based on this turbine

[3]See https://pypsa-eur.readthedocs.io for details.

specification and the recalculated wind speed at hub height, the time series is computed for all areas where wind turbines may be installed.

To derive wind capacity-factor time series at the clustered node level, PyPSA-Eur computes a weighted average of the instantaneous capacity factors $c_x(t)$ from all grid cells x within a node's Voronoi region V. The weights depend on the fractional overlap $I_{V,x}$, the mean capacity factor c_x and the maximally installable capacity $G_x^{\max}$ in each cell. This procedure ensures that cells with larger overlap, higher wind quality and greater installable potential contribute proportionally more to the aggregated node-level capacity factor time series $\bar{c}_V(t)$. [7]

$$\bar{c}_V(t) = \sum_{x \in V} \frac{I_{V,x}\, c_x\, G_x^{\max}}{\sum_{y \in V} I_{V,y}\, c_y\, G_y^{\max}}\, c_x(t) \tag{2.1}$$

Solar Thermal/PV The local instantaneous capacity factors for solar PV and solar thermal technologies are calculated from the direct and diffuse surface solar irradiance. Apart from this difference in the underlying physical inputs, the weighting and aggregation of grid cell capacity factors to the node level follow the same procedure as for the wind turbine capacity factor time series.

Hydro The hydro in-flow time series for each country c is derived by aggregating the runoff $R_x(t)$ from all grid cells $x \in \mathcal{X}(c)$, weighted by their elevation h_x:

$$G_c^H(t) = \mathcal{S}_c \sum_{x \in \mathcal{X}(c)} h_x R_x(t), \tag{2.2}$$

where the scaling factor $\mathcal{S}_c$ ensures that the annual in-flow matches historical hydro generation $E_c^{\mathsf{EIA}}(y)$:

$$\int_{\text{year } y} G_c^H(t)\,\mathrm{d}t = E_c^{\mathsf{EIA}}(y). \tag{2.3}$$

The scaled power time series $G_c^H(t)$ is then normalized by the total installed hydro capacity in country c to obtain a country-level hydro capacity-factor time series.

2.5.2. Demand Time Series

Electricity Hourly electricity load profiles are obtained from Open Power System Data,[4] which is primarily based on ENTSO-E data published on the Transparency Platform. The data is provided at the country level.

During preprocessing, electricity demand from already electrified heating is subtracted in order to allow the power-to-heat sector to be optimised independently. In addition, industrial electricity demand is removed and later redistributed to substations based on the geographic distribution of industrial facilities.[5]

The remaining national demand is allocated to all substations in the respective country using a weighted combination of two proxies: 60% proportional to the gross domestic product (GDP) within each Voronoi cell and 40% proportional to the population. These proxies serve as indicators for the spatial distribution of industrial and residential electricity demand.

Heating The heating time series are calculated with Atlite based on weather data. The daily heat demand is derived from ambient temperature and then distributed across the day using a standard daily profiles from BDEW.[6] The resulting hourly time series for the full year is subsequently normalised and scaled to match the historical annual heat consumption coming from the JRC-IDEES data base. [20]

The cooling demand is assumed to remain constant in future years and is considered to be already fully electrified.

Mobility The mobility demand is modelled as a final energy demand and is assumed to remain constant over time. The baseline demand values are taken from the JRC-IDEES database [20] for most countries. For future years, an exogenously specified vehicle fleet and the corresponding efficiency developments are applied to adjust the final energy demand. Temperature-dependent variations in vehicle efficiency are also taken into account.

2.5.3. Resources and Capacities

Power Plant Capacities The installable power plant capacities are determined for all land-restricted technologies, in particular onshore and offshore wind turbines, photovoltaic

[4]See `https://open-power-system-data.org/` for details.
[5]See `https://www.hotmaps-project.eu/` for details.
[6]See `https://github.com/oemof/demandlib` for details.

(PV) systems and solar thermal installations. For onshore wind, the usable land area is restricted based on the Corine Land Cover database [21]. All agricultural areas, forests and semi-natural areas are considered eligible, while minimum distance requirements are applied to urban and industrial land-use classes.

Offshore wind turbines can be installed within a country's Exclusive Economic Zone, with the distinction that shallow waters allow for fixed-bottom foundations, whereas deeper waters require floating platforms. All eligible areas are further restricted using the Natura 2000 database [22], which excludes protected regions. In addition to the assumed capacity density of 10 MW/km^2, an additional correction factor of 0.3 is applied to account for conflicting land uses.

A similar approach is used for PV and solar thermal technologies, with technology-specific land-use constraints. For further details, the reader is referred to the PyPSA-Eur documentation. [7]

For already installed capacities, PyPSA-Eur uses the `powerplantmatching` package [23], which provides a harmonized and matched dataset of existing power plants for the entire European power system. By incorporating unit-specific decommissioning years, the workflow ensures that the in-place power plant infrastructure is accurately represented for any modelled year.

Biomass The workflow also determines biomass and waste potentials as energy resources for all countries. The underlying data is sourced from the ENSPRESO database [24]. During preprocessing, many of the original resource categories are excluded from use in the energy system model, while the remaining categories are aggregated into three groups: waste, biogas and solid biomass. Other resources in the model are assumed to be available without physical limits, but are associated with costs and emissions that discourage excessive use.

3. Methodology and Implementation

This chapter introduces the data processing workflow developed as part of this semester project. The workflow builds on the existing PyPSA-Eur model and leverages many of the functionalities and preprocessing steps described in the previous chapter. A detailed explanation of each component of the workflow and its role within the overall modelling framework is provided.

3.1. Workflow Overview and Logic

The implemented workflow is fully self-contained within a single folder that can be directly imported into the PyPSA-Eur repository. It requires only minimal user interaction and can be executed out of the box using default settings. To ensure compatibility with both current and future versions of PyPSA-Eur, none of the original preprocessing steps are modified. Instead, a custom configuration file is included to override essential PyPSA-Eur parameters, such as weather data sources and the target year.

A second configuration file is provided to control the additional workflow developed on top of PyPSA-Eur. This file defines default values for all parameters and includes templates for technology, demand and resource data, including cost assumptions. In the remainder of this thesis, the developed workflow will be referred to as the workflow, while the original PyPSA-Eur workflow will be explicitly named the PyPSA-Eur workflow.

The design principle of the workflow is to remain as generic as possible. All required input data are converted into the specific format needed for EnergyScopeTD only at the final stages of the process. This approach allows the workflow to be adapted and expanded with additional rules and scripts, and enables the preprocessed data to be reused in other modelling frameworks. In its current form, the workflow supports only country-level data, which should be considered when interpreting results. Consequently, the processed data are most suitable for full country-level studies or European-scale analyses using one node per country.

The workflow is automated using the Snakemake framework to remain consistent with PyPSA-Eur and to enable straightforward extensions. This setup also ensures that all steps can be executed within the same environment created during the PyPSA-Eur installation, without requiring additional packages.

3.2. Workflow Structure

The complete workflow, from raw data generation to the interpretation of results, is illustrated in Figure 3.1. As outlined in the previous section, the objective of this semester project is to automate the data preprocessing for all European countries. Accordingly, the first three steps of the workflow are of particular relevance. Among these, the third step "Adaptation and conversion workflow" represents the main methodological contribution developed in this project.

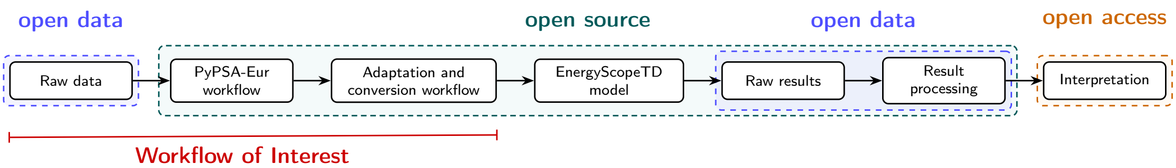

Figure 3.1.: Complete Workflow linking PyPSA-Eur and EnergyScopeTD, adapted from [12]

The three steps must be initiated individually by the user, although the first step of retrieving raw data is already included in the PyPSA-Eur workflow. However, to avoid interruptions, excessively long automated runtime, and the repeated processing of already generated data, it can be beneficial to download the preprocessed weather data separately. For this purpose, a script was developed to download the preprocessed European weather dataset from [25], which covers the years 1980 to 2020. This step is optional but significantly reduces the computational burden. Otherwise, the raw weather data must be retrieved via the Copernicus API, which is highly time-consuming.

The second step consists of running the PyPSA-Eur preprocessing workflow using the custom configuration file *config_pypsa_override.yaml*. This file allows the user to adapt the PyPSA-Eur workflow to their specific needs. In particular, it defines the weather year, the modelling horizon (final year to be modelled), and the number of clusters used to aggregate the network. It should be noted that the workflow developed in this project ultimately provides values on a per-country basis. Therefore, all nodes within each country will be merged again during subsequent processing steps. Many additional settings can be overridden in the configuration file. For a detailed description, the reader is referred to the PyPSA-Eur documentation.[1]

The final step is the workflow developed in this project, for which a detailed description is provided in the following subsections. This workflow is executed in the same way as the PyPSA-Eur workflow, using the Snakemake framework, but is controlled through its own configuration file. Figure 3.2 shows the three commands required to run the complete workflow.

```
1. python EnergyScopeTD-Eur/scripts/retrieve_weather_data.py <
   insert_year>
2. snakemake --configfile EnergyScopeTD-Eur/
   config_pypsa_override.yaml --until prepare_sector_networks --
   cores <insert_number_of_cores>
3. snakemake --snakefile EnergyScopeTD-Eur/Snakefile --cores <
   insert_number_of_cores>
```

Figure 3.2.: Commands required to execute the workflow

3.3. Workflow Implementation

As described in the previous section, the workflow is used to generate the time-series files of typical days and the corresponding data files for each European country for a selected target year, using weather data from a historical year. To implement the approach of moving from a general data structure to the specific layout required by EnergyScopeTD, the workflow is divided into several subtasks. The general structure defined in the first Snakemake rules can serve as a foundation for adding additional rules to adapt the data for other modelling frameworks.

Figure 3.3 illustrates the dependencies among the individual rules of the implemented Snakemake workflow. The following sections describe each rule and the scripts used to execute

[1] See https://pypsa-eur.readthedocs.io for details.

them. Conceptually, the workflow is divided into four components: Technologies, Resources, Demands, and Time Series.

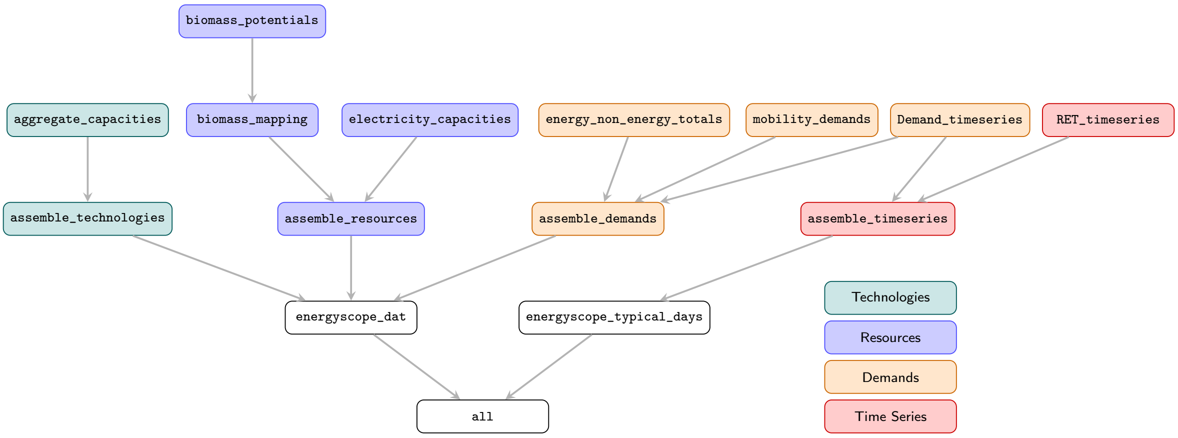

Figure 3.3.: Overview of workflow rules and dependencies.

3.3.1. Individual Steps

In this section, the individual rules and their associated scripts are presented from a high-level perspective. For details on the actual implementation, the reader is referred to the source code and the included documentation.

aggregate_capacities

In this rule, all buses from the PyPSA-Eur network file are assigned to their respective countries. Using this mapping, the installed and maximum installable capacities of all technologies are extracted for each bus. While most technologies have no explicit capacity limit, all land-restricted technologies are assigned a finite maximum potential. The script produces two intermediate CSV files: one containing the currently installed capacities per country and another containing the maximum installable capacities.

assemble_technologies

This rule uses the tables generated in *aggregate_capacities*, which contain the installed and maximum installable capacities, and integrates these values into the EnergyScopeTD technology template from the core version. The template includes additional information for all technologies, such as investment and maintenance costs, lifetime, global warming potential and availability. The output of this rule is a set of ready-to-use technology templates for all European countries.

biomass_potentials

In this rule, the biomass potentials are extracted before they are preprocessed into the PyPSA-Eur categories, as described in the previous chapter. The PyPSA-Eur preprocessing step is bypassed in order to preserve the raw potentials for conversion into the categories required for EnergyScopeTD in the subsequent steps. The raw data, aggregated for each country, are stored in an intermediate CSV file for further processing.

biomass_mapping

This rule reads the CSV file created in *biomass_potentials* and applies the aggregation according to the categories defined in the configuration file. The default categories in the core version are *waste*, *wood* and *wet biomass*. The aggregated categories per country are stored as an intermediate CSV file for further processing.

electricity_capacities

This rule reads the final network file from the PyPSA-Eur workflow and computes the international electricity exchange capacities for each country. It sums the nominal capacity of all transmission lines that connect nodes in different countries and aggregates the available exchange capacity at the country level. The results are stored in an intermediate CSV file for further processing.

assemble_resources

This rule uses the intermediate CSV files produced by *biomass_mapping* and *electricity_capacities* to assemble the resource table required for EnergyScopeTD. The values are used to populate the resource template defined in the configuration file, which contains additional information on the resources available in the core version of EnergyScopeTD, such as global warming potential and costs.

energy_non_energy_totals

This rule loads the projected energy demands in the target year for the clustered nodes in the PyPSA-Eur workflow before they are aggregated into broader categories, as well as the non-energy industrial demands, such as cement and steel. All available categories are then aggregated at the country level and stored in an intermediate CSV file for further processing.

mobility_demands

This rule extracts the raw mobility data from the databases used for PyPSA-Eur in order to avoid the assumptions applied during its preprocessing. As described earlier, PyPSA-Eur models mobility as an energy demand, whereas EnergyScope represents mobility as a service demand in person- and ton-kilometres per country. For the EU-27 countries, the database used by PyPSA-Eur already contains both person- and ton-kilometre values, which are directly extracted.

For non-EU countries, only the final energy consumption of the mobility sectors is available in the databases used for PyPSA-Eur. In this case, the energy demand is extracted and converted from ktoe into person- or ton-kilometres using the average EU efficiencies for each category. For some mobility categories, it was necessary to further disaggregate the available data into the required categories prior to conversion. This was also performed using EU average ratios. The resulting data are harmonized across all countries and stored in an intermediate CSV file for further processing.

Demand_timeseries
This rule extracts the load time series from all nodes in the network file generated by the PyPSA-Eur workflow. Based on the extracted data, the time series for each demand type are aggregated at the country level and the normalized time series are calculated. The script also computes the absolute annual demand values. The time series for each country are stored individually in intermediate CSV files, while the total annual demands for all countries are saved in a single CSV file for further processing.

assemble_demands
This rule loads the intermediate CSV files from *Demand_timeseries*, *mobility_demands* and *energy_non_energy_totals* to construct the yearly demand table required for the EnergyScopeTD data file. The script obtains the total demands for each category from *Demand_timeseries* and, if necessary, splits the individual demands across households, services, and industry using the reference ratios from *energy_non_energy_totals*. The total mobility demands from *mobility_demands* are then added. All values are finally integrated into the demand template defined in the configuration file and saved as a CSV file for further processing.

RET_timeseries
This rule extracts the capacity factor time series for all renewable technologies and aggregates them at the country level using a weighted average based on the installed or installable capacities. The resulting time series for each country are stored in individual intermediate CSV files for further processing.

assemble_timeseries
This rule reads the time series CSV files generated by *Demand_timeseries* and *RET_timeseries*

and assembles the final time series file required for EnergyScopeTD. The resulting file is saved as a CSV for each country and is used to select the typical days for the optimisation. The detailed process is described in the following section.

3.3.2. EnergyScopeTD Data File

The intermediate tables from the *technologies*, *resources*, and *demands* rules are used in the *energyscope_dat* rule to construct the final AMPL data file required to run the optimisation for each country. The script used to generate the data file was developed by Gabriel Wiest at ETH Zurich and was only adapted where necessary to match the format of the workflow.

The script uses the technology table from *assemble_technologies*, the resource table from *assemble_resources* and the demand table from *assemble_demands* for each country. These tables are used to extract all required sets, such as end-use types, technologies, resources, and sectors. The tables themselves are also included in the data file, as they are necessary to construct the final energy system representation.

In addition to the information generated in the workflow, further data is required, such as the conversion efficiencies of different technologies and exogenously defined values (e.g., an upper bound for public transport). All required information is provided in easy-to-read CSV tables and JavaScript Object Notation (JSON) files.

The output of this rule is a ready-to-use AMPL data file for each European country. These files can be used as a basis for extensive energy system analyses or refined with more detailed information provided by the user.

3.3.3. EnergyScopeTD Typical Days

The rule *energyscope_typical_days* reads the country-level time series files created in *assemble_timeseries* and selects a predefined number of representative days from the full year. Before the clustering process, the script assigns predefined weights to each time series, which can be specified by the user (the default value is one).

The core algorithm used for the selection process is implemented in AMPL, where the problem is formulated as a mixed-integer linear program (MILP). The objective function minimised is the Euclidean distance between daily time series. The method for selecting representative days was originally developed in [26]. Using this approach, the algorithm selects the optimal set of days that represent the entire year and assigns each original day to one representative day.

After the selection, the script computes scaling factors for the reduced time series of typical days to ensure that the overall energy corresponds to the full-year values. The final step is to write the results for the typical days into the AMPL format required by EnergyScopeTD. The output of this rule is a ready-to-use representation of typical days for each country, which forms the basis for the reduced time-resolution optimisation in EnergyScopeTD.

4. Results and Discussion

In this chapter, the results of the developed workflow are discussed. First, the general output of the workflow is presented, followed by a detailed evaluation of the accuracy of individual parameters. For this evaluation it is important to note that the PyPSA-Eur workflow preceded the developed workflow across most categories and that uncertainties and errors are consequently propagated through it. For this reason, special emphasis is placed on identifying results that require refinement or should be reviewed in detail when using the data. In the final part of this chapter, a small example is provided to illustrate the application of the workflow and the type of analysis it enables within a limited timeframe.

4.1. General Results

The final output of the developed workflow consists of 34 data files and 34 time series files, each corresponding to one European country. The workflow runs completely automated with minimal input from the user, who only needs to execute three commands to produce all files after optionally adjusting the configuration files, if desired. The workflow can therefore be run truly "out of the box" and represents a significant step toward the wider adoption of EnergyScopeTD, contributing to the simple-to-use scenario analysis tool envisioned by Limpens et al. (2019) [4]. In addition, the workflow stores intermediate results in a more general data format, enabling straightforward extension to other optimization frameworks. This enhances comparability across models and helps avoid duplication of work.

Together with the provided model file developed by Limpens et al. (2019) [4] for the core version of EnergyScopeTD, the generated workflow files can immediately be used to run the optimization model for a specific country. For this purpose, a Jupyter Notebook [27] is also included in the repository, allowing users to solve the energy system model and investigate the results. The notebook currently offers basic analytical capabilities, including displaying total system costs, plots of capacity factors for renewable technologies across typical days, installed capacities for all technologies and resource usage in the modeled year. An example plot of PV capacity factors is shown in Figure 4.1.

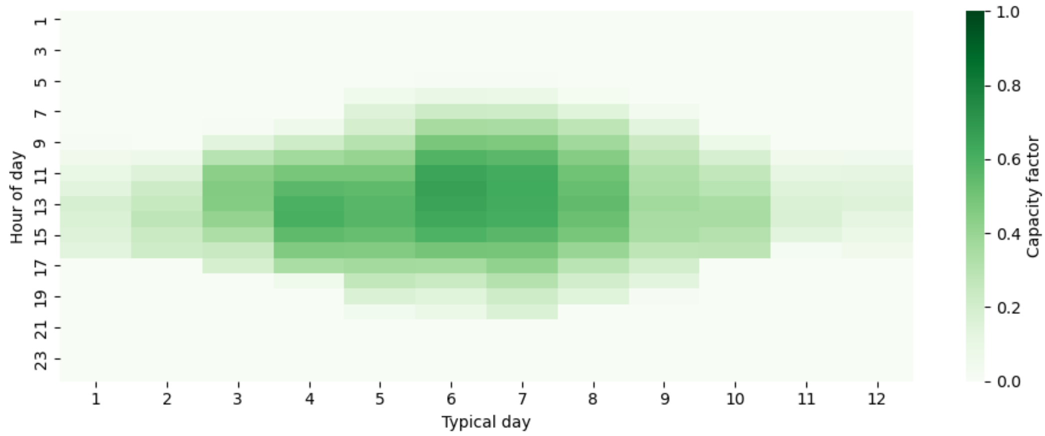

Figure 4.1.: PV Capacity Factors for Typical Days in Germany 2018

4.2. Validation

As discussed in Section 2.2, the results of energy system models are inherently difficult to validate, as they are strongly assumption-based and no ground truth for future developments can be established. For this reason, the entire workflow is implemented in an open-source framework and relies exclusively on open databases, allowing model assumptions to be transparently inspected and scrutinized. All explicit assumptions made in this project are documented in Chapter 3 and implicit assumptions were avoided wherever possible.

Although the outcomes of the optimization model itself cannot be directly validated, several input parameters, particularly those partially derived from historical data, can be assessed for plausibility and compared against observed statistics. Such an evaluation provides valuable insights and helps identify areas requiring further refinement and calibration. The following section presents this analysis for a selected set of parameters.

4.2.1. Weather dependent data

Inputs generated from weather reanalysis databases, such as ERA5 [18], can be compared with observed statistics for the corresponding year. In particular, the time series of weather dependent renewable technologies and heating demand are of high relevance, as they strongly influence the overall model behaviour.

Figure 4.2 shows the calculated time series of space heating demand for Germany based on weather data from 2015. As described in Section 2.5.2, the total annual demand is calibrated to match statistical data for the corresponding year. Consequently, validation of the annual demand levels is not required. Validation of the temporal profile, however, is considerably more challenging, as publicly available measurement data with sufficient temporal resolution are scarce.

The only publicly available dataset with comparable granularity is the gas consumption published by the United Kingdom Transmission System Operator[1]. An evaluation of this dataset in the context of the Atlite package was previously conducted by Antonini et al. (2024) [28], who reported good agreement between modelled and measured demand profiles. Since the weather dependent demand data from Atlite were not modified within the present workflow, but only normalised to conform to the EnergyScope input format, a similar level of agreement can be assumed for this project.

More generally, the correlation between degree heating days and observed heating demand, which forms the basis of the Atlite methodology, has been confirmed as an accurate predictor in numerous studies [29].

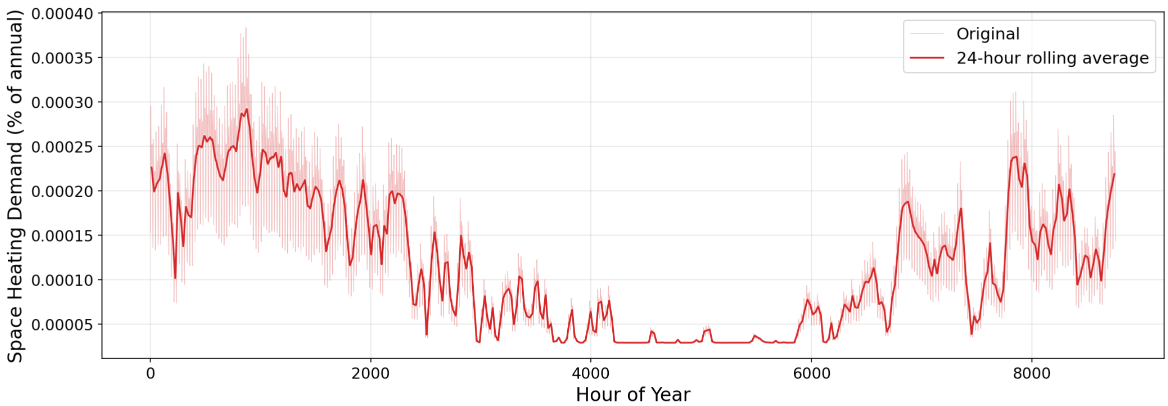

Figure 4.2.: Hourly Space Heating Demand Time Series Germany 2015

The annual production potentials of weather dependent technologies, with the exception of hydropower, are not calibrated. Table 4.2.1 presents the full load hours of weather dependent technologies for Germany in the years 2018 and 2019. For comparison, estimates

[1] See `https://data.nationalgas.com/find-gas-data` for details.

of full load hours derived from measured production data provided by Energy Charts of the Fraunhofer Institute for Solar Energy Systems (ISE)[2] are included. These values were calculated and rounded to the nearest 50 hour increment.

Since full load hour data are not directly available, the annual electricity production was divided by the installed capacity to obtain an estimate. This calculation assumes a linear installation of capacity over the course of the year and uses the installed capacity at mid year as the reference value.

Technology	2019 (Workflow)	2019 (ISE)	2018 (Workflow)	2018 (ISE)
PV	1118	1000	1171	1050
Wind Onshore	2140	1950	1931	1700
Wind Offshore	5148	3650	4906	3500
Run-of-River	4018	(4000)	3593	(3050)

Table 4.1.: Comparison FLH Germany (2018 and 2019)

The data for Germany clearly indicate that the calculated full load hours are consistently higher than those derived from measured electricity production data. The full load hours for hydropower are shown in brackets, as the ISE values also include pumped storage and reservoir hydropower. Reservoir hydropower is not included as a technology in the core version of EnergyScope, which limits the comparability with observed production data.

For the remaining technologies, several effects may explain the observed differences. One contributing factor is that theoretically calculated capacity factors do not account for downtime due to maintenance or curtailment. Curtailment alone amounted to approximately 3.5 percent in both years according to the International Energy Agency[3]. In the case of photovoltaic power, the type of installation, namely rooftop or utility scale systems, also plays an important role in assessing accuracy. These two technology types are not differentiated in either EnergyScope or the ISE dataset.

A further contributing factor is the national averaging applied within the workflow. As described in Section 2.5.1, capacity weighting is implemented under the assumption that installations are preferentially built in areas with favourable wind or solar conditions. Depending on the actual spatial distribution of installations, this approach may lead to either an overestimation or an underestimation of the resulting full load hours.

[2] See `https://www.energy-charts.info/index.html?l=de&c=DE` for details.
[3] See `https://www.iea.org/reports/renewable-energy-market-update-june-2023` for details.

Overall, the data show a reasonable level of agreement with measured values for Germany when the described uncertainties are taken into account. However, these effects are not sufficient to explain all observed deviations. Similar studies using reanalysed weather data have also identified discrepancies and concluded that they can largely be attributed to known biases in the ERA5 dataset. These biases arise from limitations such as simplified terrain orography, insufficient coverage of assimilated observations and relatively low model resolution [28].

To obtain more realistic capacity factor time series, a calibration factor should therefore be applied to reduce these systematic errors. It is important that this calibration is performed individually for each country and technology, as the underlying effects may lead to an underestimation of capacity factors in some regions and an overestimation in others. One illustrative example is Switzerland, where the calculated full load hours for onshore wind turbines are strongly underestimated at 463 h per year for 2019, while photovoltaic capacity factors are overestimated at 1378 h per year for the same year. Both the direction and the magnitude of these deviations remain relatively consistent throughout the years, indicating that the application of a constant calibration factor may be sufficient.

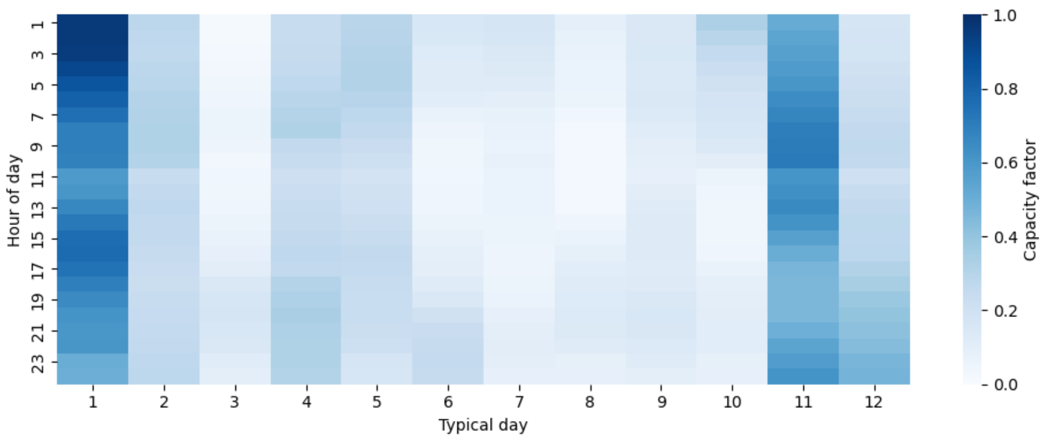

Figure 4.3.: Onshore Wind Capacity Factors for Typical Days in Germany 2018

These substantial deviations in the case of Switzerland can largely be attributed to the complex terrain, where local wind effects are not adequately represented by the coarse spatial resolution of the ERA5 dataset. A more detailed discussion and analysis of these effects is provided by Antonini et al. (2024) [28]. The study further demonstrates that the temporal evolution of capacity factors is in good agreement with measurement data, indicating that an adjustment of the absolute values by means of a calibration factor is sufficient. Figure 4.3

presents the hourly capacity factors for onshore wind in 2018, illustrating the typical increase in production at the beginning and end of the year.

4.2.2. Demands and other data

Almost all annual demand values, as well as the electricity time series, are obtained from official data sources and therefore do not require further validation. The main exceptions are the heating demand, which was discussed in the previous section, and the mobility demand for countries outside the European Union. The calculation methods for these demands are described in Chapter 3.3.1.

Mobility	2019 (Workflow)	2019 (FSO)	EU-27(JRC)
Passenger [Mpkm]	99,512	129,984	6,856,261
Freight [Mtkm]	42,089	27,972	2,534,320

Table 4.2.: Comparison Mobility Demand Switzerland in 2019

As an example, Table 4.2.2 compares the calculated passenger- and tonne-kilometres obtained from the workflow with the official statistics from the Swiss Federal Statistical Office for the base year 2019.[4] In addition, the table reports the cumulative values for the European Union in 2019 based on the JRC-IDEES database [20].

The comparison clearly shows that passenger transport is underestimated, while freight transport is overestimated in the workflow. This discrepancy can be attributed to differences in the ratio of passenger to freight transport between Switzerland and the European Union. In 2019, the ratio of passenger-kilometres to tonne-kilometres was 2.7 in the EU, compared to 4.6 in Switzerland. This effect was expected when applying EU-average values, but this approach was chosen due to the lack of sufficiently detailed country-specific data within the PyPSA-Eur framework. If higher accuracy is required, country-specific passenger-to-freight transport ratios should be researched and incorporated into the workflow.

In summary, the results demonstrate that the developed workflow is capable of generating comprehensive and internally consistent input data for EnergyScopeTD with minimal user interaction. Where validation against historical data is possible, the generated parameters show reasonable agreement with observed statistics, while remaining deviations can largely be explained by known methodological limitations and data uncertainties inherited from

[4] See `https://litra.ch/de/oev-fakten/die-litra-verkehrszahlen-2019-sind-da/` for details.

upstream workflows. The analysis highlights in particular the need for calibration of weather-dependent capacity factors. Nevertheless, the presented results confirm that the workflow provides a robust and transparent default data basis that enables out-of-the-box analyses for all European countries. At the same time, the generated data allow for targeted refinement by the modeler, depending on the intended application.

4.3. Example: Effects of Different Weather Years

This section presents an example application of the data generated by the workflow to illustrate the types of analyses facilitated by the developed approach. For this case study, the workflow was executed five times using five different weather years as the underlying data basis.

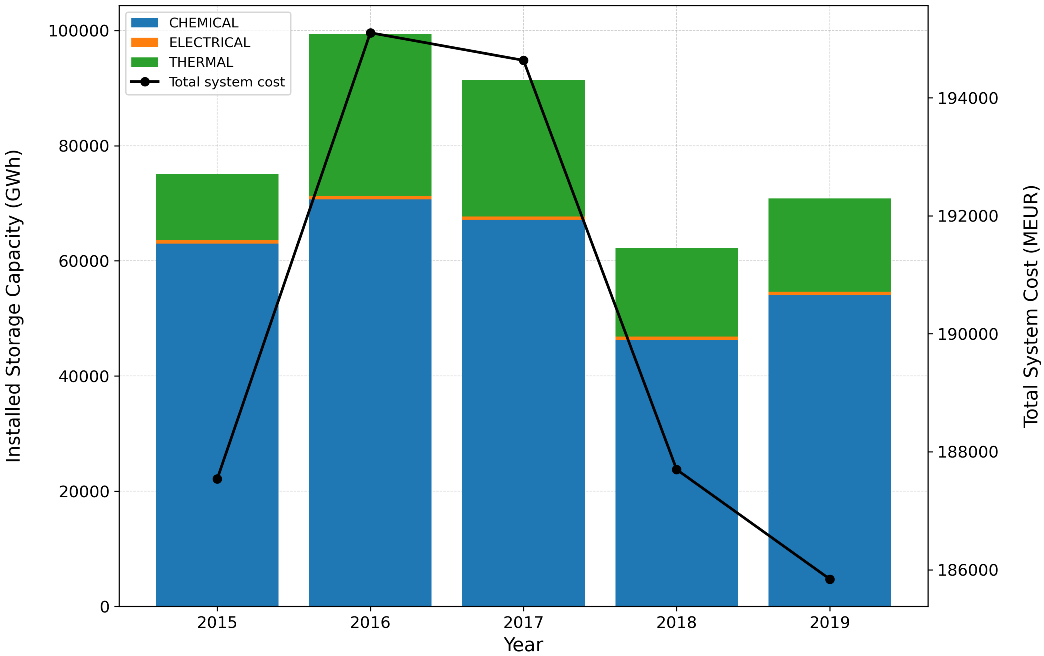

Figure 4.4.: Installed storage capacities and total system costs for a cost-minimal German energy system in 2050 under different weather years

Figure 4.4 presents how the installed storage capacities of a cost-minimal German energy system in 2050 vary with the choice of basis year. The storage capacities are grouped into chemical, electrical, and thermal storage, while the corresponding total annual system costs are indicated by the black dots.

The results reveal a clear correlation between the required storage capacities and the overall system costs. In energy systems dominated by variable renewable energy sources, larger temporal mismatches between supply and demand lead to higher storage requirements. The graph also highlights the importance of considering different weather years in energy system optimization, as they can lead to substantially different outcomes. In this example, total system costs vary by up to 5% across the considered years.

Across all scenarios, the system relies predominantly on thermal storage for short-term heat balancing, typically on daily timescales, enabling increased electricity consumption during periods of high availability. In contrast, seasonal energy storage is primarily provided by synthetically produced fuels, which allow surplus renewable electricity to be shifted across longer time horizons.

This analysis is not discussed in further detail, as it serves solely as an illustrative example of potential use cases. The analysis was significantly accelerated by the high degree of automation provided by the developed workflow. In a conventional modeling setup, a substantial amount of effort would have been required for data collection and preprocessing prior to conducting the analysis.

5. Conclusion and Outlook

- Recap
- scaling for renewables needed
- Outlook, extensions to other Frameworks, building up a universal energy model pre-processing workflow, make models comparable, open assumptions etc.

Bibliography

[1] Jing Qiu, Junhua Zhao, Fushuan Wen, Junbo Zhao, Ciwei Gao, Yue Zhou, Yuechuan Tao, and Shuying Lai. Challenges and Pathways of Low-Carbon Oriented Energy Transition and Power System Planning Strategy: A Review. *IEEE Transactions on Network Science and Engineering*, 11(6):5396–5416, November 2024. ISSN 2327-4697, 2334-329X. doi: 10.1109/TNSE.2023.3344729. URL `https://ieeexplore.ieee.org/document/10372130/`.

[2] Robert Gaugl, Kelvin Walenta, and Sonja Wogrin. A comparative analysis of energy system modeling frameworks. *e+i Elektrotechnik und Informationstechnik*, October 2025. ISSN 0932-383X, 1613-7620. doi: 10.1007/s00502-025-01345-x. URL `https://link.springer.com/10.1007/s00502-025-01345-x`.

[3] Edward Anderson, Michael Ferris, Andrew Philpott, Mihai Anitescu, Peter Cramton, Sijia Geng, Richard Green, Tito Homem-de Mello, Olivier Huber, Vincent Leclère, and Ramteen Sioshansi. Ten challenges for mathematical modeling of the green-energy transition. *Current Sustainable/Renewable Energy Reports*, 12(1): 26, September 2025. ISSN 2196-3010. doi: 10.1007/s40518-025-00274-9. URL `https://link.springer.com/10.1007/s40518-025-00274-9`.

[4] Gauthier Limpens, Stefano Moret, Hervé Jeanmart, and Francois Maréchal. EnergyScope TD: A novel open-source model for regional energy systems. *Applied Energy*, 255:113729, December 2019. ISSN 03062619. doi: 10.1016/j.apenergy.2019.113729. URL `https://linkinghub.elsevier.com/retrieve/pii/S0306261919314163`.

[5] Tom Brown, Jonas Hörsch, and David Schlachtberger. PyPSA: Python for Power System Analysis, January 2018. URL `http://arxiv.org/abs/1707.09913`. arXiv:1707.09913.

[6] Nicholas Gorman, Iain MacGill, and Anna Bruce. How to support the adoption of open-source energy system modelling software? Insights from interviews with users

and developers. *Energy Research & Social Science*, 111:103479, May 2024. ISSN 22146296. doi: 10.1016/j.erss.2024.103479. URL https://linkinghub.elsevier.com/retrieve/pii/S2214629624000707.

[7] Jonas Hörsch, Fabian Hofmann, David Schlachtberger, and Tom Brown. PyPSA-Eur: An Open Optimisation Model of the European Transmission System, October 2018. URL http://arxiv.org/abs/1806.01613. arXiv:1806.01613.

[8] Stefan Pfenninger, Adam Hawkes, and James Keirstead. Energy systems modeling for twenty-first century energy challenges. *Renewable and Sustainable Energy Reviews*, 33:74–86, May 2014. ISSN 13640321. doi: 10.1016/j.rser.2014.02.003. URL https://linkinghub.elsevier.com/retrieve/pii/S1364032114000872.

[9] Christos A. Frangopoulos. Recent developments and trends in optimization of energy systems. *Energy*, 164:1011–1020, December 2018. ISSN 03605442. doi: 10.1016/j.energy.2018.08.218. URL https://linkinghub.elsevier.com/retrieve/pii/S0360544218317547.

[10] Anthony Papavasiliou. *Optimization Models in Electricity Markets*. Cambridge University Press, 1 edition, June 2024. ISBN 9781009416627 9781009416610. doi: 10.1017/9781009416627. URL https://www.cambridge.org/highereducation/product/9781009416627/book.

[11] Stefan Pfenninger. Open code and data are not enough: understandability as design goal for energy system models. *Progress in Energy*, 6(3):033002, July 2024. ISSN 2516-1083. doi: 10.1088/2516-1083/ad371e. URL https://iopscience.iop.org/article/10.1088/2516-1083/ad371e.

[12] Stefan Pfenninger, Lion Hirth, Ingmar Schlecht, Eva Schmid, Frauke Wiese, Tom Brown, Chris Davis, Matthew Gidden, Heidi Heinrichs, Clara Heuberger, Simon Hilpert, Uwe Krien, Carsten Matke, Arjuna Nebel, Robbie Morrison, Berit Müller, Guido Pleßmann, Matthias Reeg, Jörn C. Richstein, Abhishek Shivakumar, Iain Staffell, Tim Tröndle, and Clemens Wingenbach. Opening the black box of energy modelling: Strategies and lessons learned. *Energy Strategy Reviews*, 19:63–71, January 2018. ISSN 2211467X. doi: 10.1016/j.esr.2017.12.002. URL https://linkinghub.elsevier.com/retrieve/pii/S2211467X17300809.

[13] Stefan Pfenninger, Joseph DeCarolis, Lion Hirth, Sylvain Quoilin, and Iain Staffell. The importance of open data and software: Is energy research lagging behind? *Energy Policy*, 101:211–215, February 2017. ISSN 03014215. doi: 10.1016/j.enpol.2016.11.046. URL https://linkinghub.elsevier.com/retrieve/pii/S0301421516306516.

[14] Frauke Wiese, Ingmar Schlecht, Wolf-Dieter Bunke, Clemens Gerbaulet, Lion Hirth, Martin Jahn, Friedrich Kunz, Casimir Lorenz, Jonathan Mühlenpfordt, Juliane Reimann, and Wolf-Peter Schill. Open Power System Data - Frictionless data for electricity system modelling, January 2019. URL http://arxiv.org/abs/1812.10405. arXiv:1812.10405.

[15] T. Brown, D. Schlachtberger, A. Kies, S. Schramm, and M. Greiner. Synergies of sector coupling and transmission reinforcement in a cost-optimised, highly renewable European energy system, July 2018. URL http://arxiv.org/abs/1801.05290. arXiv:1801.05290.

[16] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Peter C. Van Dyken, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Filipe G. Vieira, Christian Meesters, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Jake VanCampen, Venkat Malladi, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. Sustainable data analysis with Snakemake. *F1000Research*, 10:33, September 2025. ISSN 2046-1402. doi: 10.12688/f1000research.29032.3. URL https://f1000research.com/articles/10-33/v3.

[17] Fabian Hofmann, Johannes Hampp, Fabian Neumann, Tom Brown, and Jonas Hörsch. atlite: A Lightweight Python Package for Calculating Renewable Power Potentials and Time Series. *Journal of Open Source Software*, 6(62):3294, June 2021. ISSN 2475-9066. doi: 10.21105/joss.03294. URL https://joss.theoj.org/papers/10.21105/joss.03294.

[18] C3S. ERA5 hourly data on single levels from 1940 to present, 2018. URL https://cds.climate.copernicus.eu/doi/10.24381/cds.adbb2d47.

[19] Uwe Pfeifroth, Steffen Kothe, Jaqueline Drücke, Jörg Trentmann, Marc Schröder, Nathalie Selbach, and Rainer Hollmann. Surface Radiation Data Set - Heliosat (SARAH) - Edition 3, May 2023. URL https://wui.cmsaf.eu/safira/action/viewDoiDetails?acronym=SARAH_V003.

[20] Mate Rozsai, Marc Jaxa-Rozen, Raffaele Salvucci, Przemyslaw Sikora, Jacopo Tattini, and Frederik Neuwahl. JRC-IDEES-2021. May 2024. URL http://data.europa.eu/89h/82322924-506a-4c9a-8532-2bdd30d69bf5.

[21] European Environment Agency. CORINE Land Cover 2012 (vector), Europe, 6-yearly - version 2020_20u1, May 2020, 2019. URL https://sdi.eea.europa.eu/catalogue/copernicus/api/records/916c0ee7-9711-4996-9876-95ea45ce1d27?language=all.

[22] European Environment Agency and European Commission. Natura 2000 (vector) - version 2022, 2024. URL https://sdi.eea.europa.eu/catalogue/srv/api/records/95e717d4-81dc-415d-a8f0-fecdf7e686b0?language=all.

[23] Fabian Gotzens, Heidi Heinrichs, Jonas Hörsch, and Fabian Hofmann. Performing energy modelling exercises in a transparent way - The issue of data quality in power plant databases. *Energy Strategy Reviews*, 23:1–12, January 2019. ISSN 2211467X. doi: 10.1016/j.esr.2018.11.004. URL https://linkinghub.elsevier.com/retrieve/pii/S2211467X18301056.

[24] CASTELLO Pablo Ruiz, Wouter Nijs, Dalius Tarvydas, Alessandra Sgobbi, Andreas Zucker, Roberto Pilli, Andrea Camia, Christian Thiel, Carsten Hoyer-Klick, LONGA Francesco Dalla, Tom Kober, Jake Badger, Patrick Volker, Berien Elbersen, Andre Brosowski, Daniela Thrän, and Klas Jonsson. ENSPRESO - an open data, EU-28 wide, transparent and coherent database of wind, solar and biomass energy potentials, 2019. URL https://publications.jrc.ec.europa.eu/repository/handle/JRC116900.

[25] Aleksander Grochowicz and Koen van Greevenbroek. ERA5 Weather Data Cutouts for PyPSA-Eur (1980-2020), 2022. URL https://archive.sigma2.no//dataset/2346FE44-17CC-49EF-96B2-F9850B14E13D.

[26] Fernando Domínguez-Muñoz, José M. Cejudo-López, Antonio Carrillo-Andrés, and Manuel Gallardo-Salazar. Selection of typical demand days for CHP optimization. *Energy and Buildings*, 43(11):3036–3043, November 2011. ISSN 03787788. doi: 10.1016/j.enbuild.2011.07.024. URL https://linkinghub.elsevier.com/retrieve/pii/S037877881100329X.

[27] Brian E. Granger and Fernando Perez. Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering*, 23(2):7–14, March 2021. ISSN 1521-9615, 1558-366X. doi: 10.1109/MCSE.2021.3059263. URL `https://ieeexplore.ieee.org/document/9387490/`.

[28] Enrico G. A. Antonini, Alice Di Bella, Iacopo Savelli, Laurent Drouet, and Massimo Tavoni. Weather- and climate-driven power supply and demand time series for power and energy system analyses. *Scientific Data*, 11(1):1324, December 2024. ISSN 2052-4463. doi: 10.1038/s41597-024-04129-8. URL `https://www.nature.com/articles/s41597-024-04129-8`.

[29] Iain Staffell, Stefan Pfenninger, and Nathan Johnson. A global model of hourly space heating and cooling demand at multiple spatial scales. *Nature Energy*, 8(12):1328–1344, September 2023. ISSN 2058-7546. doi: 10.1038/s41560-023-01341-5. URL `https://www.nature.com/articles/s41560-023-01341-5`.

A. Formula

Formula