SoftwareX 32 (2025) 102393

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

UrbanHeatOpt: A software framework for supporting municipal heat transition planning

S. Malacek [a,*], D. Cardona-Vasquez [a], J. Portela [b], I. Abbate [c], Y. Werner [a], S. Wogrin [a]

[a] *Institute of Electricity Economics and Energy Innovation at Graz University of Technology, Inffeldgasse 18, Graz, Austria*
[b] *Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas, Calle Alberto Aguilera 23, Madrid, Spain*
[c] *Innomerics, Segundo Mata, 4, 28224, Pozuelo de Alarcón, Madrid, Spain*

ARTICLE INFO

Keywords:
Energy system modeling
District heating
Time series generation
Urban energy planning
Waste heat utilization

ABSTRACT

A transition in the heating sector – particularly for buildings – is essential to address climate change. Heating solutions will need to vary significantly depending on local conditions such as climate, building standards, and the availability of waste heat. Emerging generations of low-temperature district heating systems enable the integration of new waste heat sources, but identifying feasible, initial design heating concepts remains a time-consuming task. In this paper, we present `UrbanHeatOpt`, a user-friendly, modular software framework that streamlines the entire process — from generating high-resolution heat demand data to identifying potential community heating concepts. The tool supports both early-stage planning in municipalities and more in-depth analyses to inform policy-making in the heating sector.

Code metadata

Current code version	v1.0.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-25-00504
Permanent link to Reproducible Capsule	–
Legal Code License	MIT Licence
Code versioning system used	git
Software code languages, tools, and services used	Python environment v. 3.12
Compilation requirements, operating environments & dependencies	Mini-conda installation
If available Link to developer documentation/manual	https://iee-tugraz.github.io/UrbanHeatOpt/
Support email for questions	simon.malacek@tugraz.at

1. Motivation and significance

Advanced optimization models [1,2] and comprehensive datasets for transnational generation [3] and transmission expansion planning [4] are already available to support decision-making in the power sector. In contrast, the heating sector still lags behind and remains significantly more fragmented. Regional differences in climate conditions, building standards, heating technologies, ownership structures, and urban morphology hinder the application of universal solutions. Consequently, effective heat transition strategies must be developed at the district or municipal level [5]. Although there are existing tools for detailed building simulations [6] or district heating network design (e.g., detailed in energyPRO [7], or more holistic in nPro [8]), these are often not suited for early-stage planning. Many require highly granular input data and domain-specific expertise, limiting their usability in exploratory or policy-driven scenarios.

Research and practical planning for a sustainable, just and cost-effective heat transition are constrained by several persistent challenges:

- **Data scarcity:** Detailed, high-resolution data on building-level thermal demand is often unavailable, particularly for entire urban districts. Public datasets may lack temporal granularity by only providing yearly demand.
- **Scalability and user behavior:** Most building simulation tools provide high-fidelity results for individual buildings but are difficult to scale to hundreds or thousands of units. Additionally,

---

* Corresponding author.
E-mail address: simon.malacek@tugraz.at (S. Malacek).

https://doi.org/10.1016/j.softx.2025.102393
Received 18 July 2025; Received in revised form 22 September 2025; Accepted 30 September 2025

they often omit stochastic variations among buildings introduced by occupant behavior.
- **Complex workflows:** Existing workflows typically require manual preprocessing, expert parameter selection, and multiple software environments. This creates a barrier to accessibility beyond experts, especially in municipal planning or interdisciplinary research contexts.

To address these limitations, we present an open and modular software framework that enables quick and easy-to-use feasibility studies with light data requirements for community-scale heating concepts. The tool is designed for both early-stage municipal planning and scientific experimentation. Its key features include:

1. **Estimation of annual thermal demand and building properties** using publicly available geospatial data.
2. **Generation of stochastic, behavior-aware demand time series**, incorporating user occupancy patterns, daily routines, and weather sensitivity.
3. **Integrated spatial clustering**, allowing computationally efficient aggregation of demand across urban subregions while preserving temporal fidelity.
4. **Linear optimization modeling** for district heating network layout, technology investment planning, and optimal use of waste heat and renewable sources.

Compared to existing high-fidelity tools (e.g., TRNSYS [9]), the proposed framework shifts the focus from detailed, resource-intensive simulation towards scalable and customizable exploration of future sustainable heating concepts through optimization under limited data availability. However, expanding into these new application areas also entails trade-offs, such as relying on simplified representations of thermal energy storage (TES) and heat flows within the district heating network. The modular architecture and transparency allow researchers to isolate and reuse individual components (e.g., heat demand time series synthesis or optimization routines). Planners, on the other hand, can deploy the full pipeline to generate initial designs in a matter of hours. In this way, the tool serves two distinct but complementary user groups:

- **Municipal Planners:** The software supports the early stages of community heating plans, such as those required by recent national policies [10]. By requiring minimal expert input and offering sensible default parameters, it enables local authorities to evaluate alternative scenarios and prioritize further analysis quickly.
- **Researchers:** The open, parameterized architecture facilitates investigations into the effects of climate conditions, retrofit strategies, internal heat gains, and policy mechanisms. Researchers can rely on a validated data pipeline and simulation framework, freeing them to focus on model extensions or hypothesis testing.

The full pipeline is implemented in Python, with each function documented for standalone use. A Jupyter Notebook guides users through the entire low-code workflow – from data acquisition to final visualization – while allowing intermediate data access for custom analyses. Users can adjust input parameters to test sensitivities or tailor assumptions to local contexts.

A comprehensive software description is provided in Section 2, followed by an illustrative example in Section 3. Section 4 discusses broader implications, and Section 5 concludes with a future outlook.

2. Software description

`UrbanHeatOpt` is developed to accelerate early-stage investigations of integrated heating concepts at the district or community level. In contrast to traditional physical building simulation tools, which require detailed inputs for each building, our software estimates building-level heat demand based on geospatial building data, standardized typologies [11], and publicly available metadata.

A key feature is the generation of realistic, synthetic heat demand time series that reflect both behavioral occupancy patterns and external environmental conditions presented in [12]. Thanks to its modular design, alternative approaches can also be integrated, such as those presented in [13]. These demand profiles serve as inputs for a subsequent techno-economic optimization module. Within this module, users can define potential waste heat sources, as well as investment candidates for TES and power-to-heat (P2H) technologies.

The core of the software is a mixed-integer linear program (MILP) model that determines the cost-optimal combination of technology investments, storage strategies, and a potential district heating network. While it does not model detailed hydraulic or thermal behavior, the approach provides significantly more insight than static heating density methods (e.g., [14]) that are commonly used in spatial energy planning.

Section 2.1 describes the software architecture and workflow, while Section 2.2 explains selected algorithmic and technical aspects in more detail. Finally, Section 2.3 elaborates on the computational performance of the software.

2.1. Software architecture

`UrbanHeatOpt` is implemented in Python and designed for execution within a Jupyter Notebook or implementation in other scripts. A conda environment specification is provided to manage all dependencies and ensure compatibility. Details can be found in the User Manual [15].

Fig. 1 presents the overall folder structure. Geospatial data is stored in GeoJSON format to preserve geometries and projection metadata, while time series are saved as CSV files.

The software distinguishes between *case studies* and *scenarios*: A case study defines a specific geographical location with associated building stock data, while different scenarios can be defined to test varying cost assumptions, policies, or technology configurations.

Each analysis is initiated through the Jupyter Notebook `main.ipynb` (Fig. 2), which coordinates the workflow and allows users to customize key parameters. Table 1 provides an overview of input types and sources.

The main workflow consists of five sequential steps, displayed in Fig. 3:

1. **Geodata Preparation** (`prepare_geodata.py`): Building data is extracted from OpenStreetMap (OSM) using the `OSMnx` [21] package. Missing attributes, such as floor number or construction year, are estimated via heuristics or default values. Buildings are then matched to archetypes using parameters in `BuildingTypology.xlsx`. Yearly heat demand is calculated and exported to `Building_Data.geojson`.
2. **Time Series Generation** (`hd_time_series_generator.py`): A Markov Chain Monte Carlo method generates stochastic active occupancy profiles for each dwelling. These drive a thermal model to compute hourly heating demand, considering weather and internal gains [12]. Final time series are stored in `Building_TS.csv`.
3. **Clustering and Network Setup** (`clustering.py`): To reduce computational complexity, buildings are grouped and a potential district heating network is proposed. The network and all investment candidates are saved in `Heat_Network.geojson`.
4. **Optimization** (`model.py`): A MILP model implemented in `Pyomo` [22,23] identifies the cost-optimal configuration of technologies, storage, and network investments over a typical year. Results are saved in the scenario's `output` and `expost` subfolders.

```
project_root/
├── default/                        # Default input templates used for scenario configuration
│   ├── input_ParameterCosts.xlsx          # Default cost parameters (e.g. piping, investment)
│   ├── input_HeatGenerationUnits.xlsx     # Default waste heat unit definitions
│   └── …
├── Conda-Activation-Scripts/       # Activation script(s) for setting up the Python environment
├── docs/                           # Sphinx documentation source files
├── logs/                           # Log files from optimization model runs
├── [case_study_name]/              # Top-level folder for one spatial region (case study)
│   ├── general_parameters/         # Static settings specific to this case study
│   │   ├── building_energy_data/           # Heat demand by building type and structure
│   │   └── MCMC_data/                      # Transition matrices, weather, and solar data
│   ├── Building_Data.geojson              # Spatial data for the region (from OSM or other sources)
│   ├── Building_TS.csv                    # Generated heat demand time series
│   └── scenarios/                                 # directory for all scenarios
│       ├── [scenario_name]/            # One or more scenarios for analysis within a case study
│       │   ├── data/                          # Scenario-ready data for optimization
│       │   ├── input/                         # User-defined, scenario-specific parameters
│       │   │   ├── input_HeatGenerationUnits.xlsx  # Waste heat source definitions
│       │   │   ├── input_ParameterCosts.xlsx       # Cost and design parameters
│       │   │   └── input_WastHeatProfiles.xlsx     # Temporal profiles for waste heat
│       │   ├── output/                        # Raw output files from the optimization model
│       │   └── expost/                        # Post-processing and analysis results
│       └── …
├── _config.yaml                     # Central configuration for file paths and defaults
├── environment.yml                 # Conda environment requirements
├── main.ipynb                      # Main notebook to run the full data workflow
├── prepare_geodata.py              # Module: Prepare and download geospatial data
├── hd_time_series_generator.py     # Module: Generate stochastic heat demand time series
├── clustering.py                   # Module: Cluster demand and propose network layouts
├── model.py                        # Module: Optimization model for energy system configuration
├── data.py                         # Module: Utilities for data handling and preprocessing
├── utils.py                        # Module: General helper functions
└── visualisation.py                # Module: Visual output and plotting
```

Fig. 1. Overview of the folder structure containing required files and scripts.

Table 1
Main input data and corresponding sources. Different data sources can be used to account for specific local considerations. File and folder names refer to the flowchart in Fig. 3 and file structure in Fig. 1.

Description	File	Folder	Source
Building information	–	Web/API	OpenStreetMap [16]
Building typology	BuildingTypology.xlsx	building_energy_data	TABULA WebTool [11]
Outdoor temperature	outside_temp.xlsx	MCMC_data	Renewables.ninja [17,18]
Solar gains	solar_gain.xlsx	MCMC_data	Renewables.ninja [17,18]
Active occupancy patterns	transition_matrix_WD.xlsx, transition_matrix_WE.xlsx	MCMC_data MCMC_data	Time Use Survey [19]
General parameters and costs	input_ParameterCosts.xlsx	input	User-defined
Heat generation units	input_HeatGenerationUnits.xlsx	input	PETA [20]
Waste heat profiles	input_WastHeatProfiles.xlsx	input	User-defined

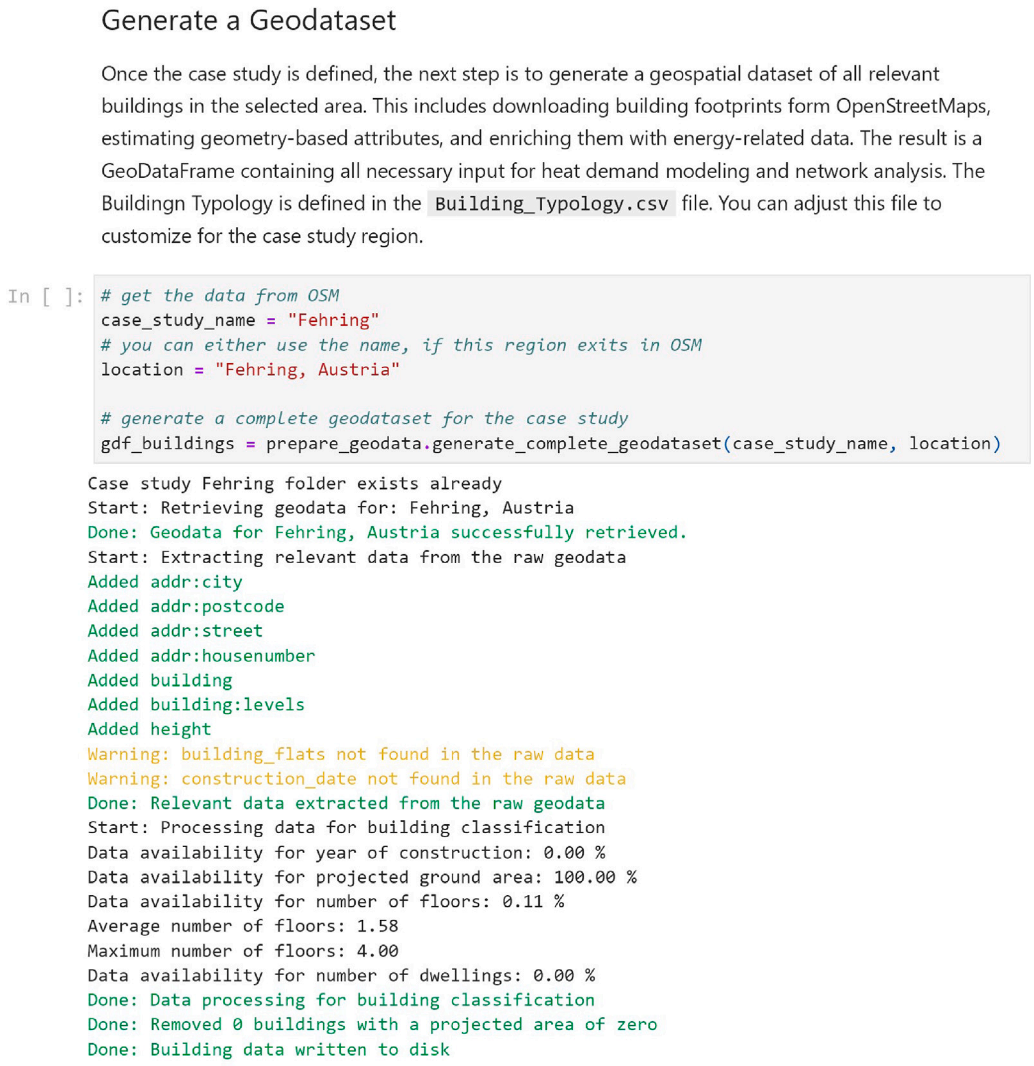
5. **Visualization** (`visualisation.py`): Standard result visualizations are generated and stored in the `plots` folder.

A User Manual [15], generated using the `Sphinx` [24] package, provides information on setting up and running the tool, as well as a description of all inputs and options. Inline docstrings within the code further offer complete documentation for each function, enabling easy reuse of individual components. Details of the mathematical formulation and conceptualization are provided in the Supplementary Material [25].

2.2. Software functionalities

Data preprocessing and fallback heuristics. OSM data quality can vary significantly depending on the region. In some cases, attributes such as the number of floors, roof shape, or construction year may be missing. To address this, the software implements fallback heuristics and default values to ensure a consistent and robust input dataset. For example, if the number of floors is not available, it is estimated based on the building's footprint area and assumed base-length-to-height ratio. This approach improves automation and reproducibility across different case studies.

Time series generation and computational acceleration. The generation of hourly heating demand time series for each building is computationally demanding. In contrast to simple degree-day [26] or load-profile methods [27], the implemented approach combines active occupancy profiles with weather data to dynamically compute space heating demands. Since the demand at a given hour depends on the state at the previous timestep (i.e., internal building temperature), vectorization across time is not possible. To achieve fast runtimes for large datasets (e.g., several thousand buildings with 8760 timesteps), the implementation uses `NumPy` [28] arrays and the `Numba` [29] package to precompile time-critical loops to machine code. This results in



Generate a Geodataset

Once the case study is defined, the next step is to generate a geospatial dataset of all relevant buildings in the selected area. This includes downloading building footprints form OpenStreetMaps, estimating geometry-based attributes, and enriching them with energy-related data. The result is a GeoDataFrame containing all necessary input for heat demand modeling and network analysis. The Buildingn Typology is defined in the `Building_Typology.csv` file. You can adjust this file to customize for the case study region.

```
# get the data from OSM
case_study_name = "Fehring"
# you can either use the name, if this region exits in OSM
location = "Fehring, Austria"

# generate a complete geodataset for the case study
gdf_buildings = prepare_geodata.generate_complete_geodataset(case_study_name, location)
```

```
Case study Fehring folder exists already
Start: Retrieving geodata for: Fehring, Austria
Done: Geodata for Fehring, Austria successfully retrieved.
Start: Extracting relevant data from the raw geodata
Added addr:city
Added addr:postcode
Added addr:street
Added addr:housenumber
Added building
Added building:levels
Added height
Warning: building_flats not found in the raw data
Warning: construction_date not found in the raw data
Done: Relevant data extracted from the raw geodata
Start: Processing data for building classification
Data availability for year of construction: 0.00 %
Data availability for projected ground area: 100.00 %
Data availability for number of floors: 0.11 %
Average number of floors: 1.58
Maximum number of floors: 4.00
Data availability for number of dwellings: 0.00 %
Done: Data processing for building classification
Done: Removed 0 buildings with a projected area of zero
Done: Building data written to disk
```

Fig. 2. Screenshot of the Jupyter Notebook showing part of the workflow for gathering data from OpenStreetMap for a specific location.

speed-ups of more than an order of magnitude compared to native Python.

Flexible clustering and heating network setup. A typical case study contains hundreds or even thousands of individual buildings. To reduce the computational burden of the subsequent optimization problem, the software performs a clustering step in which spatially proximate buildings are aggregated. This is achieved using the k-means algorithm from the `scikit-learn` [30] package, with the number of clusters defined by the user. Within each cluster, demand time series are aggregated. By varying the number of clusters, users can balance computational efficiency and modeling accuracy — using fewer clusters for rapid initial design iterations and more clusters for detailed, high-accuracy analyses in later stages. To propose a potential district heating network topology, a Delaunay triangulation using `SciPy` [31] of the cluster centroids and potential heat sources is computed. Based on that, a minimum spanning tree is derived using the `NetworkX` [32] package. While this approach does not guarantee a perfect network configuration, it provides a plausible, first estimate of possible network connections that can later be refined.

Optimization model implementation. The optimization problem is formulated as a MILP using `Pyomo` [22,23]. The objective function minimizes the total annual system cost, including investment and operating costs for central and decentralized (i.e., on-site heating systems) heat generation, P2H conversion, and TES units. The model enforces energy balance constraints, technology-specific capacity limits, and temporal storage dynamics. The implementation enables users to adjust all technological parameters and costs, as well as define new generation or TES units via input files. The complete mathematical formulation can be found in the Supplementary Material [25].

2.3. Computational performance

Table 2 presents the computation times for the processing steps in the case study described in Section 3, indicating the computational effort. The calculations were performed on a standard laptop (Windows 11, Intel i7-1360P, 32 GB RAM). The execution time of all the steps – except for the building and solving of the optimization model – scales approximately linearly with the number of buildings included. The solving time of the optimization model strongly depends on the specific input and the solver used. Currently, `gurobi` [33] and `HiGHS` [34] solvers are implemented.

3. Illustrative examples

To showcase the main features of `UrbanHeatOpt`, we demonstrate a case study for the community of *Fehring*, a town with approximately 7000 inhabitants in Styria, Austria. Starting with only publicly available data, we aim to investigate: (i) the approximate heating demand at high spatial and temporal resolution, (ii) the potential feasibility of

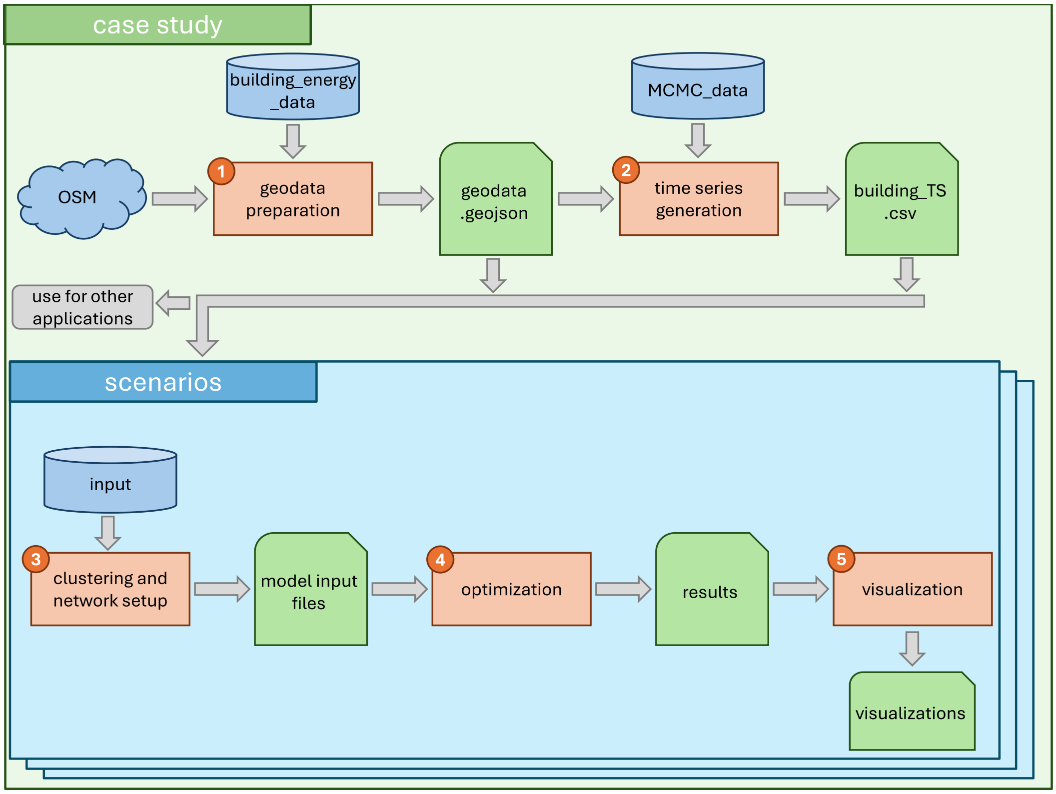


Fig. 3. Flowchart of the complete data pipeline: data sources (folder name according to Table 1) are shown in blue, software modules (as enumerated in Section 2.1) in orange, and intermediate and final outputs in green.

Table 2
Computation times for processing steps for a case study of 16 937 dwellings in 8760 time steps.

Processing step	Computation time / s
1. Geodata Preparation	3
2. Time Series Generation	39
3. Clustering and Network Setup	7
4. Optimization - model building	185
- solving	approx. 1800
5. Visualization	5

using waste heat or central heating units, and (iii) where a local district heating network could be viable.

The analysis was conducted using the Jupyter Notebook interface (see Fig. 2). The first step involves accessing building data from OSM. By specifying the location `"Fehring, Austria"`, the script retrieves 6211 buildings of various types. The command line output also includes a data quality summary — e.g., indicating that this OSM dataset contains limited information about building heights. If more detailed data (e.g., from a cadastre) is available, it can be easily merged in based on location or postal address.

A folder for the case study is automatically created, where building and weather data can be adapted for the specific location using publicly available sources. The total annual heating demand for this region is estimated at 92 GW h. Fig. 4 shows an interactive visualization of the spatially resolved heating demand in the notebook.

Next, we run the `time series generation` script, which creates a stochastic, weather- and solar-irradiance-dependent heat demand time series for each building. The resulting aggregated time series is shown as the black demand curve in Fig. 6.

We then include potential waste heat sources. From the Styrian heat atlas [35], we identify three sources: one industrial waste heat source at intermediate temperature with 10.4 MW capacity and 6000 full-load hours, and two sources from wastewater treatment plants, with 220 kW and 60 kW continuous potential. These could be used directly or with heat pumps [36]. We also add one TES unit and one heat boiler as investment candidates next to the industrial waste heat source. Standard cost and performance parameters are used. For details on the implementation of heat sources, see the Supplementary Material [25].

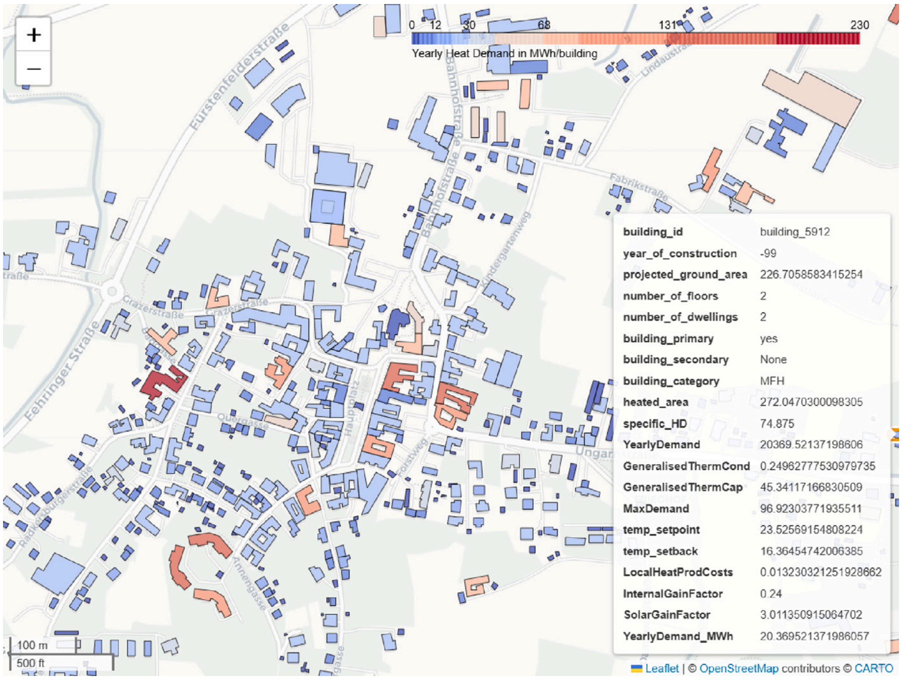
After clustering and setting up the network, we run the optimization model. Using the built-in visualization module, we generate Figs. 5 and 6. The former illustrates investment decisions in waste heat utilization, heat generation, and thermal storage. The latter shows hourly operational results for the case study, including demand and supply profiles.

To run different case studies, the case study folder can simply be copied, renamed, and adapted by changing parameters – for example, to reflect different costs or investment candidates – allowing for the investigation of various configurations.

In conclusion, this quick analysis reveals a limited but promising opportunity to utilize 6 MW of the industrial waste heat within a localized network in the town center. The system includes a TES unit of $6960\,\mathrm{m}^3$ for load shifting and a backup boiler with a capacity of 2 MW, designed based on the given cost assumptions and parameters. These initial results, combined with the generated high-resolution heat demand data, offer a strong foundation for a more in-depth and detailed technical planning.

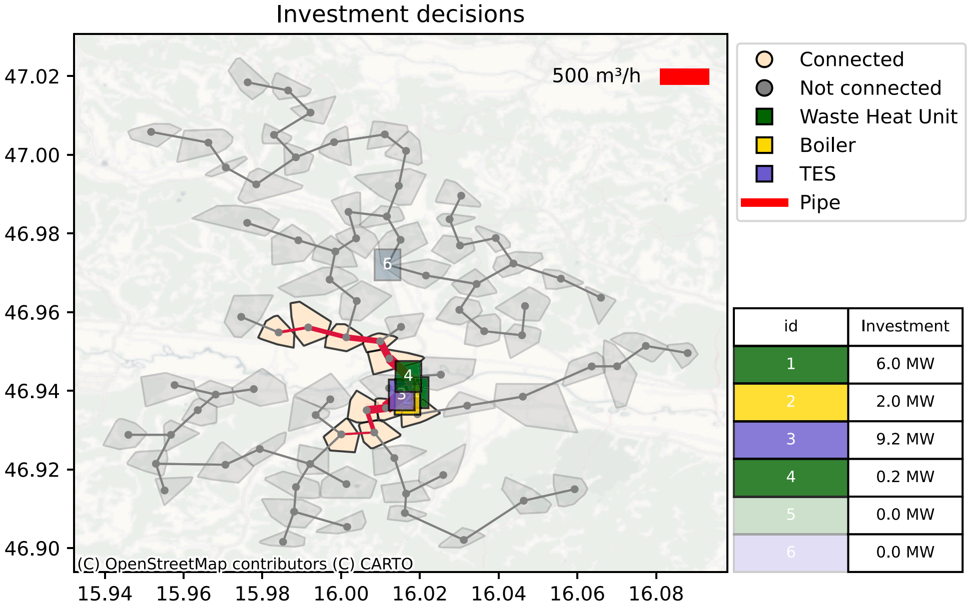
4. Impact

The software package presented here addresses key challenges in (i) energy system modeling, planning, and policy evaluation by providing high-quality heat demand data, (ii) enabling optimal heat infrastructure design, and (iii) supporting equitable and accelerated heat transition strategies.



building_id	building_5912
year_of_construction	-99
projected_ground_area	226.7058583415254
number_of_floors	2
number_of_dwellings	2
building_primary	yes
building_secondary	None
building_category	MFH
heated_area	272.0470300098305
specific_HD	74.875
YearlyDemand	20369.52137198606
GeneralisedThermCond	0.24962777530979735
GeneralisedThermCap	45.34117166830509
MaxDemand	96.92303771935511
temp_setpoint	23.52569154808224
temp_setback	16.36454742006385
LocalHeatProdCosts	0.013230321251928662
InternalGainFactor	0.24
SolarGainFactor	3.011350915064702
YearlyDemand_MWh	20.369521371986057

Fig. 4. Interactive visualization of building-level heating demand in a part of Fehring.



id	Investment
1	6.0 MW
2	2.0 MW
3	9.2 MW
4	0.2 MW
5	0.0 MW
6	0.0 MW

Fig. 5. Investment decisions in district heating system infrastructure.

Data is a key element in energy system modeling across all scales. While electricity demand is often backed by measured data from substations or smart meters, heat demand remains significantly more challenging to quantify. Simply aggregating standard load profiles tends to produce unrealistic load peaks [27]. The `geodata preparation` and `time series generation` module of this software package addresses this issue by delivering realistic, geo-localized heat demand data. By ensuring correlation with provided outdoor temperature and solar irradiation profiles, the module becomes a convenient and reliable data source for a wide range of energy modeling applications — thus improving model accuracy. It serves a role similar to that of renewables.ninja [17], which has become a standard source for PV and wind capacity factor time series. A preliminary version of this module has already been used in [12] to compare heat demand derived from cadastral data with that produced by the open-source method introduced in this work.

Historically, waste heat integration in district heating has focused on cogeneration power plants and high-temperature industrial processes. However, the ongoing shift toward low-temperature heat networks (4th- and 5th-generation district heating [37]) enables the utilization of alternative waste heat sources, such as supermarket refrigeration systems, electrical substations, wastewater, data centers [38], and electrolyzers [39]. These developments open up opportunities for novel and more innovative district heating system configurations, while also

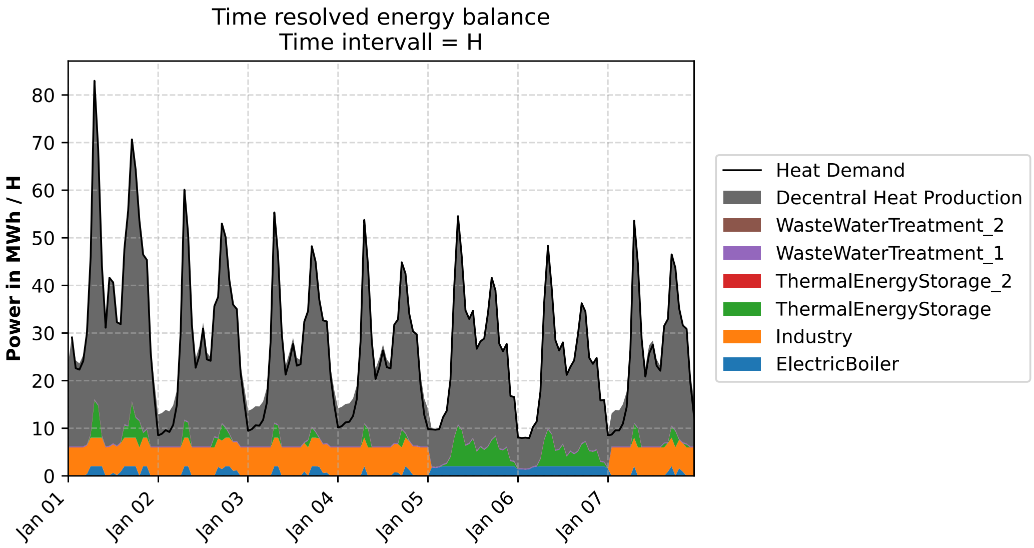


Fig. 6. Operational results: hourly heat demand and supply for the Fehring case study.

calling for new approaches to system design and planning. The `optimization` module of this software allows for rapid identification of optimal system designs, including waste heat integration and network layout. For example, it was applied in the *Atmosphere* project [40,41] to assess the feasibility of using waste heat from an electrolyzer in Puertollano, Spain. Through its publicly available, open-data approach, this tool supports the evaluation of district heating concepts not only for researchers but also for policymakers, planners, and community stakeholders.

Finally, the ongoing heat transition raises critical questions about how to accelerate and ensure a just transition [42]. Due to the decentralized nature of building and heating system ownership, the transformation will require numerous private investments in insulation and retrofitting. These will only proceed at the necessary pace [43] if supported by suitable policy measures, subsidies, affordable alternatives, or publicly funded infrastructure such as new district heating networks. This software package can support the analysis and evaluation of such measures in early planning stages. In this way, the software can support the technical and economic planning of heating concepts at the municipal level (as required, for example, in Germany [10]), even by users without detailed expert knowledge.

5. Conclusions

In this paper, we present the novel open and modular software `UrbanHeatOpt`, designed to generate high-resolution heat demand data and support the investigation of optimal community heat system configurations. Its modular architecture offers flexible functionality, enabling easy integration into other software tools or customization for specific research applications. As demonstrated in the case study, the software allows for the low-effort development of reasonable initial heating concepts at the community level, making it a valuable tool for real-world planning. At the same time, it provides a solid foundation for in-depth analysis of targeted policy questions. We welcome contributions and suggestions from the wider heat energy system community – both researchers and practitioners – to improve and expand this open-source tool.

CRediT authorship contribution statement

S. Malacek: Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **D. Cardona-Vasquez:** Software, Methodology. **J. Portela:** Supervision, Project administration. **I. Abbate:** Software. **Y. Werner:** Writing – review & editing, Supervision. **S. Wogrin:** Writing – review & editing, Supervision, Resources.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT from OpenAI in order to enhance readability and perform grammar and spell-checking. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jose Portela reports financial support was provided by Centre for the Development of Industrial Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Many thanks to Felix C. A. Auer for his invaluable assistance with coding and for generously sharing his extensive knowledge in successfully managing software projects.

This research was supported by funding from the Spanish Centre for the Development of Industrial Technology (CDTI), with Grant Number MIG-20221006 associated with the ATMOSPHERE Project.

Supported by TU Graz Open Access Publishing Fund.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.softx.2025.102393.

Data availability

Default data for running the software, including the dataset of the test case study presented in Section 3, is available in the code repository.

References

[1] Wogrin S, Tejada-Arango DA, Gaugl R, Klatzer T, Bachhiesl U. LEGO: The open-source Low-carbon Expansion Generation Optimization model. SoftwareX 2022;19:101141. http://dx.doi.org/10.1016/j.softx.2022.101141.

[2] Hörsch J, Hofmann F, Schlachtberger D, Brown T. PyPSA-Eur: An open optimisation model of the European transmission system. Energy Strat Rev 2018;22:207–15. http://dx.doi.org/10.1016/j.esr.2018.08.012.

[3] Koltsaklis NE, Dagoumas AS. State-of-the-art generation expansion planning: A review. Appl Energy 2018;230:563–89. http://dx.doi.org/10.1016/j.apenergy.2018.08.087.

[4] Lumbreras S, Ramos A. The new challenges to transmission expansion planning. survey of recent practice and literature review. Electr Power Syst Res 2016;134:19–29. http://dx.doi.org/10.1016/j.epsr.2015.10.013.

[5] Herreras Martínez S, Harmsen R, Menkveld M, Faaij A, Kramer GJ. Municipalities as key actors in the heat transition to decarbonise buildings: Experiences from local planning and implementation in a learning context. Energy Policy 2022;169:113169. http://dx.doi.org/10.1016/j.enpol.2022.113169.

[6] Pan Y, Zhu M, Lv Y, Yang Y, Liang Y, et al. Building energy simulation and its application for building performance optimization: A review of methods, tools, and case studies. Adv Appl Energy 2023;10:100135. http://dx.doi.org/10.1016/j.adapen.2023.100135.

[7] Østergaard PA, Andersen AN, Sorknæs P. The business-economic energy system modelling tool energyPRO. Energy 2022;257:124792. http://dx.doi.org/10.1016/j.energy.2022.124792.

[8] Wirtz M. nPro: A web-based planning tool for designing district energy systems and thermal networks. Energy 2023;268:126575. http://dx.doi.org/10.1016/j.energy.2022.126575.

[9] TRNSYS: Transient System Simulation Tool. https://www.trnsys.com/.

[10] Gesetz für die Wärmeplanung und zur Dekarbonisierung der Wärmenetze. https://www.bmwsb.bund.de/SharedDocs/gesetzgebungsverfahren/DE/kommunale-waermeplanung.html.

[11] Loga T, Stein B, Diefenbach N. TABULA building typologies in 20 European countries — Making energy-related features of residential building stocks comparable. Energy Build 2016;132:4–12. http://dx.doi.org/10.1016/j.enbuild.2016.06.094.

[12] Malacek S, Portela J, Werner Y, Wogrin S. Generating building-level heat demand time series by combining occupancy simulations and thermal modeling. Smart Energy 2025;18:100181. http://dx.doi.org/10.1016/j.segy.2025.100181.

[13] Hachez J, Chicherin S, Chaudhry AM, Bram S. Regression versus probabilistic approach for operational data – Heat demand of buildings to be connected to a district heating system. Energy Build 2023;294:113209. http://dx.doi.org/10.1016/j.enbuild.2023.113209.

[14] Persson U, Werner S. Heat distribution and the future competitiveness of district heating. Appl Energy 2011;88(3):568–76. http://dx.doi.org/10.1016/j.apenergy.2010.09.020.

[15] Malacek S, Cardona D, Portela J. Welcome to UrbanHeatOpt! — UrbanHeatOpt 1.0.0 documentation. https://iee-tugraz.github.io/UrbanHeatOpt/.

[16] OpenStreetMap contributors. OpenStreetMap contributors Planet dump. 2017, retrieved from https://planet.osm.org, published: https://www.openstreetmap.org.

[17] Renewables.ninja. https://www.renewables.ninja/.

[18] Pfenninger S, Staffell I. Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data. Energy 2016;114:1251–65. http://dx.doi.org/10.1016/j.energy.2016.08.060.

[19] Harmonised European time use surveys - Microdata - Eurostat..

[20] Flensburg Halmstad, Aalborg Universities. Pan-european thermal atlas version 5.2. 2022, https://heatroadmap.eu/peta5/. [Accessed 01 September 2025].

[21] Boeing G. Modeling and Analyzing Urban Networks and Amenities With OSMnx. Geogr Anal 2025. http://dx.doi.org/10.1111/gean.70009.

[22] Hart WE, Watson J-P, Woodruff DL. Pyomo: modeling and solving mathematical programs in Python. Math Program Comput 2011;3(3):219–60. http://dx.doi.org/10.1007/s12532-011-0026-8.

[23] Bynum ML, Hackebeil GA, Hart WE, Laird CD, Nicholson BL, et al. Pyomo — Optimization Modeling in Python, 67 of Springer Optimization and its Applications. Cham: Springer International Publishing; 2021, http://dx.doi.org/10.1007/978-3-030-68928-5.

[24] Sphinx authors — Sphinx documentation. https://www.sphinx-doc.org/en/master/authors.html.

[25] Malacek S, Cardona-Vasquez D, Portela J, Abbate I, Werner Y, Wogrin S. Supplementary material to UrbanHeatOpt: A software framework for supporting municipal heat transition planning. 2025, http://dx.doi.org/10.3217/7ep0n-27377.

[26] Park S, Shim J, Song D. Issues in calculation of balance-point temperatures for heating degree-days for the development of building-energy policy. Renew Sustain Energy Rev 2021;135:110211. http://dx.doi.org/10.1016/j.rser.2020.110211.

[27] Büttner C, Amme J, Endres J, Malla A, Schachler B, Cußmann I. Open modeling of electricity and heat demand curves for all residential buildings in Germany. Energy Informatics 2022;5(1):21. http://dx.doi.org/10.1186/s42162-022-00201-y.

[28] Harris CR, Millman KJ, Van Der Walt SJ, Gommers R, Virtanen P, et al. Array programming with NumPy. Nature 2020;585(7825):357–62. http://dx.doi.org/10.1038/s41586-020-2649-2.

[29] Lam SK, Pitrou A, Seibert S. Numba: a LLVM-based Python JIT compiler. In: Proceedings of the second workshop on the LLVM compiler infrastructure in HPC, LLVM ’15. New York, NY, USA: Association for Computing Machinery; 2015, p. 1–6. http://dx.doi.org/10.1145/2833157.2833162.

[30] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res 2011;12(85):2825–30, http://jmlr.org/papers/v12/pedregosa11a.html.

[31] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 2020;17(3):261–72. http://dx.doi.org/10.1038/s41592-019-0686-2.

[32] Hagberg Aric A, Schult Daniel A, Swart Pieter J. Citing — NetworkX 2.1 documentation. https://networkx.org/documentation/networkx-2.1/citing.html.

[33] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2024, https://www.gurobi.com.

[34] Huangfu Q, Hall JAJ. Parallelizing the dual revised simplex method. Math Program Comput 2018;10(1):119–42. http://dx.doi.org/10.1007/s12532-017-0130-5.

[35] Land Steiermark. Digitaler Atlas Steiermark. https://gis.stmk.gv.at/wgportal/atlasmobile/map/Basiskarten/Basiskarte.

[36] Đurević D, Balić D, Franković B. Wastewater heat utilization through heat pumps: The case study of City of Rijeka. J Clean Prod 2019;231:207–13. http://dx.doi.org/10.1016/j.jclepro.2019.05.235.

[37] Lund H, Østergaard PA, Nielsen TB, Werner S, Thorsen JE, et al. Perspectives on fourth and fifth generation district heating. Energy 2021;227:120520. http://dx.doi.org/10.1016/j.energy.2021.120520.

[38] Davies G, Lagoeiro H, Turnell H, Wegner M, Foster A, et al. Evaluation of low temperature waste heat as a low carbon heat resource in the UK. Appl Therm Eng 2023;235:121283. http://dx.doi.org/10.1016/j.applthermaleng.2023.121283.

[39] Van der Roest E, Bol R, Fens T, van Wijk A. Utilisation of waste heat from PEM electrolysers – Unlocking local optimisation. Int J Hydrog Energy 2023;48(72):27872–91. http://dx.doi.org/10.1016/j.ijhydene.2023.03.374.

[40] Iberdrola SA. Atmosphere Project Information. https://www.iberdrola.com/press-room/news/detail/we-lead-the-atmosphere-consortium-to-boost-green-hydrogen-production.

[41] European Investment Bank. Iberdrola Green Hydrogen. https://www.eib.org/en/projects/pipelines/all/20210187.

[42] McKenna C, Gronlund C, Vaishnav P. Heating with justice: Barriers and solutions to a just energy transition in cold climates. Resour Conserv Recycl 2024;208:107699. http://dx.doi.org/10.1016/j.resconrec.2024.107699.

[43] Directive (EU) 2024/1275 of the European Parliament and of the Council of 24 April 2024 on the energy performance of buildings (recast) (Text with EEA relevance). 2024, http://data.europa.eu/eli/dir/2024/1275/oj/eng.