ECom4Future
Fiware Driven ENERGY COMmunities for the FUTURE

DELIVERABLE

3.2 Final Combined Report for Energy Analytics

Version 1.1

Adil Mukhtar, TU Wien

Gerald Schweiger, TU Wien

David Steen, Chalmers

Sanjay Somanath, Chalmers

Alexander Hollberg, Chalmers

Elena Malakhatka, Chalmers

Radu Plamanescu, UPB

Irene Hafner, DWH

Kilian Meusburger, DWH

Christof Sumereder, FH Joanneum

Theresa Kohl, DiLT Analytics

Christoph Siegl, DiLT Analytics

Marwa Maghnie, RWTH Aachen University

Laura Block, RWTH Aachen University

Internal Reference

Deliverable No.:	D 3.2 (2026)
Deliverable Name:	**Final Combined Report for Energy Analytics**
Lead Participant:	Vienna University of Technology (TU Wien)
Work Package No.:	3
Task No. & Name:	
Document (File):	
Issue (Save) Date:	

Document Status

	Date	Person(s)	Organisation
Author(s)			
Verification by			
Approval by			
Approval by			

Document Sensitivity

☐	**Not Sensitive**	Contains only factual or background information; contains no new or additional analysis, recommendations or policy-relevant statements
☒	**Moderately Sensitive**	Contains some analysis or interpretation of results; contains no recommendations or policy-relevant statements
☐	**Sensitive**	Contains analysis or interpretation of results with policy-relevance and/or recommendations or policy-relevant statements
☐	**Highly Sensitive Confidential**	Contains significant analysis or interpretation of results with major policy-relevance or implications, contains extensive recommendations or policy-relevant statements, and/or contain policy-prescriptive statements. This sensitivity requires SB decision.

Table of Contents

1 Introduction

This deliverable report presents the comprehensive outcomes of Work Package 3 (WP3) within the ECom4Future project. WP3 is divided into four specialized subpackages and adapts a broad range of technological avenues and practical use cases for energy communities. ECom4Future aims to tackle the inherent challenges posed by renewable energy sources within local energy communities and beyond. It seeks to provide solutions that promote long-term sustainable communities through a systematic examination of economic, technological, and social factors, including peer-to-peer trading, fault detection and diagnosis, energy forecasting, and load optimization strategies, etc. Although the primary goal of ECom4Future is to drive engagement through these advanced services, a significant challenge persists: synthetically generating power profiles, such as electricity consumption, PV generation profiles etc. Techniques such as energy forecasting, fault detection and diagnosis (FDD), and clustering analysis require extensive data, but in practice, obtaining real-world data from energy communities is often challenging, if not

impossible, due to legal restrictions and the substantial effort needed for data collection and maintenance.

2 Synthetic data generation (Chalmers)

Synthetic data refers to artificially generated data that is created using computer systems to mimic the characteristics of real-world datasets. This approach is particularly beneficial in scenarios where real data is scarce, inaccessible due to privacy concerns, or incomplete. Synthetic data can be categorized into three primary types:

Fully Synthetic Data: Generated entirely through models and does not include any real-world data in its final form. Algorithms based on statistical or machine learning models are used to replicate the distributions and characteristics of the original dataset.

Partially Synthetic Data: Combines real and synthetic data by replacing specific sensitive attributes in the real dataset with synthetic counterparts.

Anonymized Data: Real data that has been modified to obscure or remove identifiable information while maintaining its utility.

Synthetic data generation finds numerous applications in energy communities, such as modeling energy consumption patterns, simulating peer-to-peer energy trading, and enhancing demand-side management strategies. It provides a practical and scalable solution to overcome the limitations associated with accessing real-world datasets.

the process of generating synthetic data involves a systematic sequence of steps. It begins with Defining Objectives, where the purpose and specific requirements for synthetic data generation are established. This is followed by the Data Collection and Preparation phase, where real data is gathered, cleaned, and structured to inform the generation process. Next, Model Selection involves choosing appropriate statistical, agent-based, or AI-driven models tailored to the specific objectives. The Data Generation phase applies the selected model to create synthetic datasets that mimic the properties of real-world data. Finally, Validation and Feedback ensure the generated data meets the defined objectives through iterative refinement, comparing synthetic datasets against real-world benchmarks to ensure accuracy and usability. This structured approach ensures the synthetic data is robust, reliable, and aligned with the intended applications.

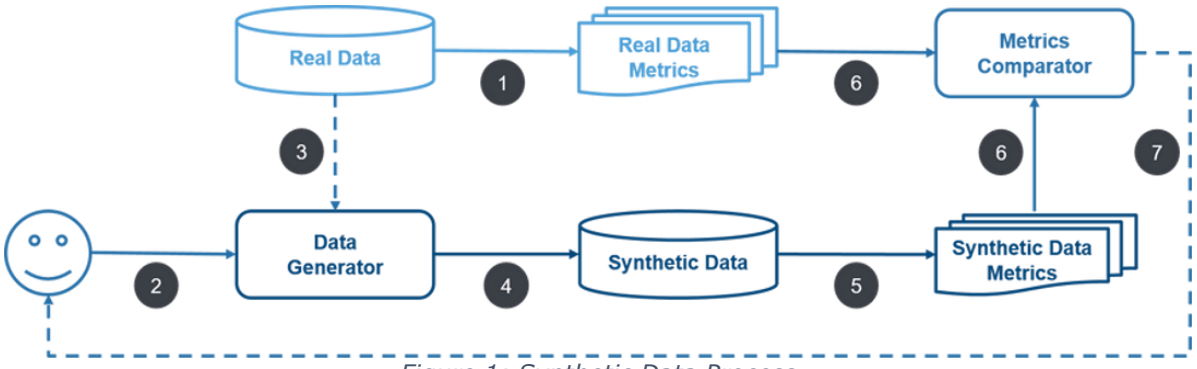

Figure 1: Synthetic Data Process

2.1 Technical Framework

Synthetic data generation methods fall into three broad categories, each with distinct methodologies and applications:

2.1.1 Model-based

Model-based approaches rely on mathematical or statistical models to replicate the behavior of systems or datasets. These methods use predefined equations or statistical distributions

to simulate data, ensuring consistency with known operational rules and trends. Example applications include simulating energy load profiles or predicting the impact of varying market structures on local energy communities.

Examples:

- In energy analytics, load profile generation using RichardsonPy is a prominent example, where stochastic models simulate realistic electricity consumption data at the household level by inputting variables such as family size, activity schedules, and weather conditions
- Fault detection in building systems can use synthetic sensor data modeled with equations from HVAC control logic.

Model-based methods are robust in scenarios with well-defined system behaviors but may require significant domain knowledge to set up accurate initial parameters.

2.1.2 Agent-based

Agent-based models focus on simulating the interactions of autonomous agents (e.g., households, devices) within a defined system. These methods are particularly effective for understanding complex systems, such as energy trading within local energy communities. Each agent operates based on predefined rules and adapts dynamically to the environment or other agents.

Examples:

- Peer-to-peer energy trading within local energy communities is often modeled using ABM, where households act as independent agents negotiating energy transactions based on supply, demand, and price conditions.
- Evaluations of distributed grid impacts from community-level renewable energy installations also benefit from this method.

ABM provides insights into emergent behaviors in complex systems and is particularly suited for decentralized energy systems, where dynamic interactions between agents significantly influence outcomes.

2.1.3 AI-based

AI-based methods leverage machine learning algorithms, such as generative adversarial networks (GANs) or variational autoencoders (VAEs), to create data that captures intricate patterns and dependencies within real datasets. These approaches are ideal for applications requiring high fidelity and adaptability, such as forecasting renewable energy outputs or detecting faults in building energy systems.

Examples:

- GANs can simulate detailed time-series data for energy consumption and production, capturing variations influenced by seasons, events, and random disturbances.
- AI models like VAEs are used to generate fault scenarios in building systems, providing labeled datasets for training fault detection algorithms.
- Physics-informed AI, which incorporates system dynamics and constraints, enhances the realism of generated data in cases like air handling unit (AHU) diagnostics.

  AI-based methods are powerful for generating high-fidelity data and can simulate datasets with complex dependencies that traditional methods struggle to reproduce.

2.2 Use Cases, Requirement Analysis and Parametric Scope

Synthetic data generation serves diverse purposes across energy systems, particularly in scenarios where real-world data is unavailable or incomplete. This section outlines key use

cases, the associated requirements, and the parametric scope relevant to synthetic data applications in energy analytics. While the implementation of several real-world use cases relies on synthetic data, this report also details specific use cases where access to actual data is possible within the project's legal and contractual framework.

2.2.1 Energy Flexibility (Chalmers)

Modeling energy flexibility within local energy communities (LECs) involves predicting consumption patterns, load profiles, and flexibility potential under various market and operational conditions.

Requirements

- Granular data: Synthetic data at a temporal resolution of 5 minutes to 1 hour.
- Scalability: Ability to simulate household-level or device-level consumption to analyze grid impacts and flexibility.
- Scenario testing: Evaluate impacts of varying incentives, market structures, and renewable energy penetration.

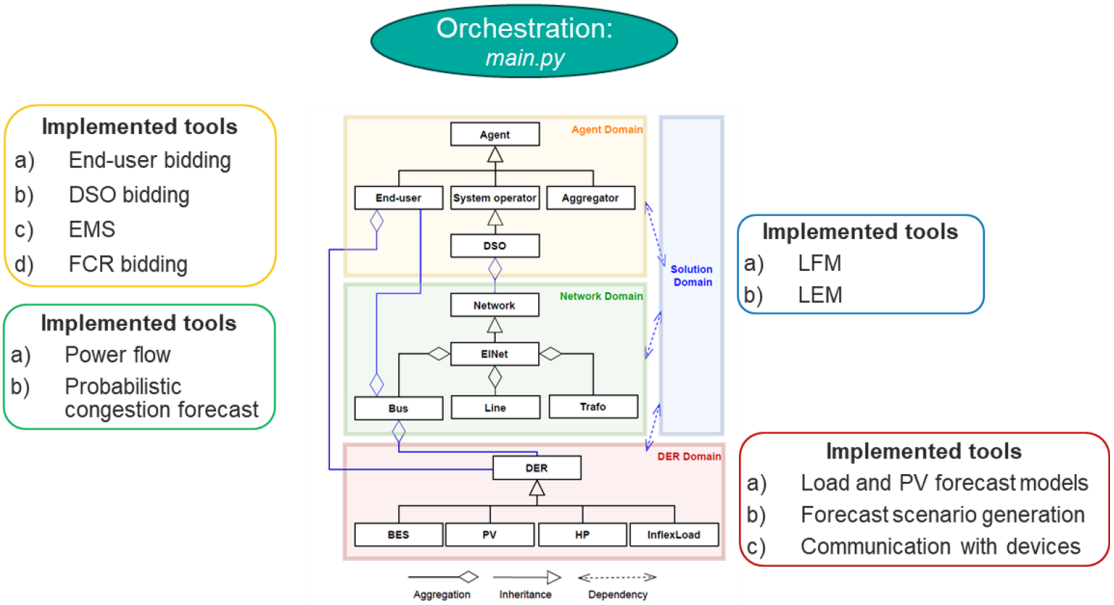

Figure 2: Local Flexibility market ABM (Chalmers use case)

Parametric Scope

- Variables: Active and reactive power, generation capacity, consumption peaks.
- Spatial resolution: Household and device-level granularity.
- Unit of measurement: Active power (kW), reactive power (kVAR).
- Tools: ABM, AI-based models such as GANs to simulate household energy profiles

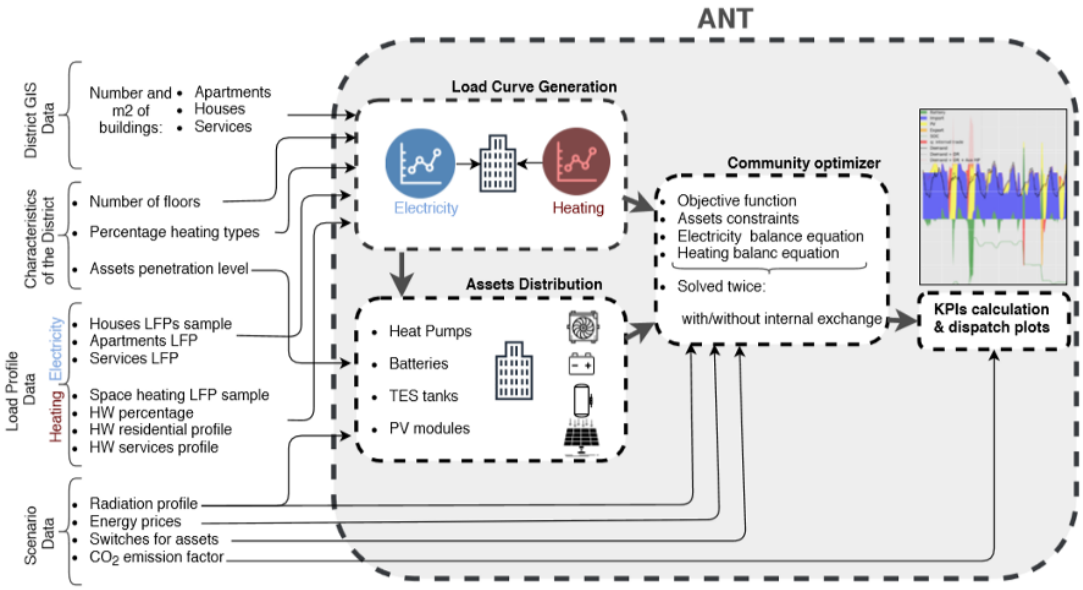

Figure 3: ANT model (Chalmers use case)

2.2.2 Forecasting, Device Profile (FH Joanneum)

Energy / power consumption forecasting for households and commonly used devices is a crucial area of research that aims to predict and optimize residential electricity usage. This field has gained significant importance due to the increasing demand for energy efficiency and the rise of smart home technologies.

Key Aspects of Household Energy and Power Consumption Forecasting, Data Collection and Analysis, accurate forecasting relies heavily on comprehensive data collection. This includes:

- Historical energy consumption data
- Lifestyle data of household occupants
- Environmental factors (temperature, humidity, etc.)
- Appliance-specific usage patterns

The most realistic way to collect this data is through measurement, whereby the 15-minute average of the smart meter has established itself as the standard. Energy consumption and power requirement profiles can be created on the basis of real measurement data and given user profiles as well as weather forecasts (temperature, sunshine duration, etc.). Recently, artificial intelligence methods have also become established:

- Machine Learning Models: Algorithms such as Support Vector Machines (SVM), Random Forests, and Neural Networks are commonly used.
- Deep Learning Approaches: Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) have shown promising results in time-series forecasting
- Hybrid Models: Combining multiple techniques, like ARIMA-SVM hybrid models, can improve prediction accuracy

Granularity of Forecasting, predictions can be made at various levels:

- Household-level: Overall energy consumption of the entire home
- Device-level: Consumption forecasts for individual appliances

- Time-scale: Predictions can range from hourly to annual forecasts

Applications and Benefits, accurate energy consumption forecasting offers several advantages:

- Demand Response Programs: Enables better management of energy supply and demand of energy storage
- Cost Savings: Helps households optimize their energy usage and reduce bills
- Smart Home Integration: Facilitates the development of intelligent energy management systems
- Grid Management: Assists utility companies in load balancing and infrastructure planning

Challenges and Future Directions, while significant progress has been made, several challenges remain:

- Data Privacy: Collecting detailed household data raises privacy concerns
- Model Interpretability: Improving the explainability of complex machine learning models
- Real-time Forecasting: Developing models capable of ultra-short-term predictions
- Integration of Multiple Data Sources: Incorporating diverse data types for more accurate forecasts

2.2.3 FDD --- Fault Detection and Diagnosis

2.2.3.1 Rule based hierarchical FDD (RWTH)

To ensure efficient operation within energy communities, robust fault detection and diagnosis (FDD) is essential. Reliable building operation is the prerequisite for grid stability, demand-side management, and local decarbonization. For such applications, detailed sensor data is fundamental. While Machine Learning approaches typically require large, labeled datasets containing both nominal and faulty states, such high-quality real-world data is often scarce or unavailable for specific systems. Consequently, our initial research plan involved generating synthetic data via simulation to train the necessary ML models. However, as the project progressed, a sufficient volume of high-quality operational data became available, rendering synthetic data unnecessary. Leveraging this real-world dataset, we implemented a rule-based hierarchical approach. This methodology bypasses the common bottleneck of missing labels while enabling precise FDD in an operational context. Chapter 8 details the Use Case and results.

Requirements:

- High temporal resolution: Fault dynamics in AHUs require data sampled at intervals of 1 minute or 15 minutes.

Parametric Scope:

- Variables: Air and water temperature, flow rates, valve openings, operating modes, setpoints.
- Timeframes: Short-term faults (hours) and long-term behaviors (weeks, months).
- Tool: Rule-based fault detection and diagnosis framework.

Use Case:

- Fault detection and diagnosis for an air handling unit (AHU)
- Measurement Devices: Sensors for air and water temperatures, valve openings, pump speed etc.

- Resolution: 5 minutes

2.2.3.2 Cluster Ensembles Learning Framework for FDD in HVAC Systems (TU Wien)

Machine learning (data-driven) methods are primarily used for Fault Detection and Diagnosis (FDD) in various automated systems. It is their ability to learn complex and non-linear patterns across a wide variety of input features that makes them highly effective. In this project, we proposed a machine learning approach for FDD in HVAC systems. HVAC systems are mainly responsible for ensuring occupants' thermal comfort and optimizing energy consumption.

We utilized classical machine learning methods as a more streamlined and robust alternative to physics based methods as they require extensive calibration for individual building geometries and struggle with real-world sensor noise. This classical data-driven approach naturally overlaps with physical principles because the input features themselves, such as temperature differentials, mass flow rates, and pressure drops, are direct reflections of the underlying thermodynamics.

Requirements:

- Access to continuous time-series data (e.g., 5 to 15-minute intervals) from key HVAC sensors, including temperature, mass flow rates, and power consumption.
- Pre-processing raw sensor inputs into physically meaningful features (such as temperature differentials and pressure drops) to ensure the classical ML models implicitly capture thermodynamic laws.
- Availability of historical data representing both nominal (healthy) operating conditions and known fault profiles to train and validate the classical ML algorithms.

Parametric Scope:

- The boundary of the system is restricted to primary air handling units (AHUs), chillers, and heating circuits, focusing on components with the highest impact on energy and comfort.
- The scope is limited to the most frequent and critical operational faults, specifically sensor drift, stuck actuator valves, and fouling in heat exchangers.

Goals/Objectives & Use Case:

- FDD for HVAC systems
- Achieve a fault detection rate greater than 90% while keeping false alarm rates below 5% to maintain building operator trust.
- Replace rigid, hard-to-calibrate physics-based simulations with a scalable, lightweight classical ML framework that reduces deployment time per building by over 40%.

2.3 Synthetic Data for Energy Analytics (Chalmers)

The ongoing electrification of society, combined with increasing shares of intermittent renewable energy, is creating a growing need for high-resolution household electricity load profiles. Applications include distribution grid planning, demand response analysis, photovoltaic and battery studies, electric vehicle (EV) charging analysis, and the design of future energy communities. In many of these applications, not only the total annual energy consumption but also the temporal structure of demand is important, particularly the coincidence of loads during peak hours.

Access to representative household electricity data is, however, often limited. Smart-meter datasets are difficult to obtain due to privacy regulations and are typically restricted in geographic coverage, household diversity, and technology penetration. Furthermore, when analysing future scenarios such as increased EV adoption or new residential developments, suitable measurement data may not yet exist. Traditional statistical approaches used in

distribution grid planning provide aggregate estimates but lack the temporal resolution and behavioural realism required for modern flexibility and peak-demand studies.

To address these challenges, the project developed SweLoadSim, a bottom-up stochastic simulator for generating synthetic Swedish household electricity load profiles with a 10-minute temporal resolution. Rather than directly modelling electricity demand, the simulator models household occupant behaviour and maps activities to appliance-level loads. The objective is to produce profiles that are individually plausible, statistically representative at population level, and sufficiently flexible to analyse future scenarios involving changing technologies, behaviours, and electrification patterns.

The simulator covers major residential electricity end uses, including lighting, cooking, media consumption, laundry, dishwashing, domestic hot water, space heating, and EV charging. Different building types, household compositions, and heating systems can be represented, allowing analysis of both apartments and single-family houses under Swedish conditions.

2.3.1 Methodology

SweLoadSim uses an activity-driven simulation framework inspired by Hägerstrand's time-geographic theory, where daily human activities are constrained by biological needs, institutional schedules, and social norms. The simulation process consists of four main stages: household generation, activity generation, activity scheduling, and activity-to-load mapping.

In the first stage, synthetic households are generated with stochastic variation in dwelling type, floor area, thermal properties, heating systems, and occupant composition. Occupants are assigned behavioural archetypes such as workers, students, retirees, or work-from-home individuals. Building parameters are varied probabilistically to capture diversity in the Swedish building stock.

In the second stage, each occupant receives a set of daily activities based on Swedish time-use survey data. Activities such as cooking, television viewing, laundry, and computer use are generated using participation probabilities and stochastic duration distributions. Household-level activities are scaled to avoid unrealistic duplication in larger households.

Figure 4 illustrates the temporal probability curves for each activity category on a typical weekday, showing how the three scheduling layers interact.

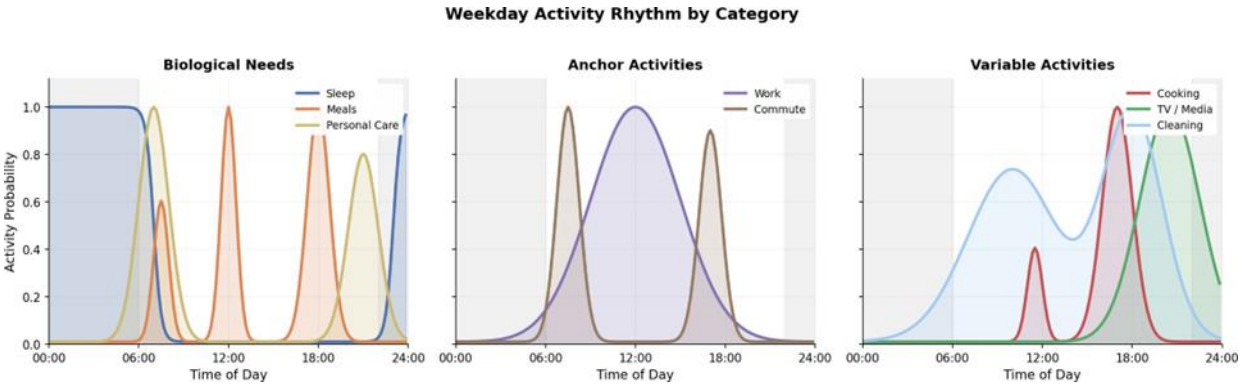

Figure 4: Temporal probability curves for weekday activities, organised by scheduling layer. Biological needs (left) show sleep dominance with meal peaks. Anchoring activities (centre) show the work-commute envelope. Flexible activities (right) fill residual time with evening-peaked patterns. Source: Gaussian curves fitted to SCB TUS 2010 diary data.

The third stage applies a hierarchical constraint-based scheduler. Biological necessities such as sleep are allocated first, followed by mandatory activities such as work and school. Flexible activities are then inserted into the remaining time slots according to empirically derived temporal probability distributions. This approach ensures realistic daily routines and avoids implausible activity sequences that may arise in

unconstrained stochastic models. Figure 5 shows a representative output: a multi-agent Gantt chart for a four-member family (two employed adults, two school-age children) on a single weekday.

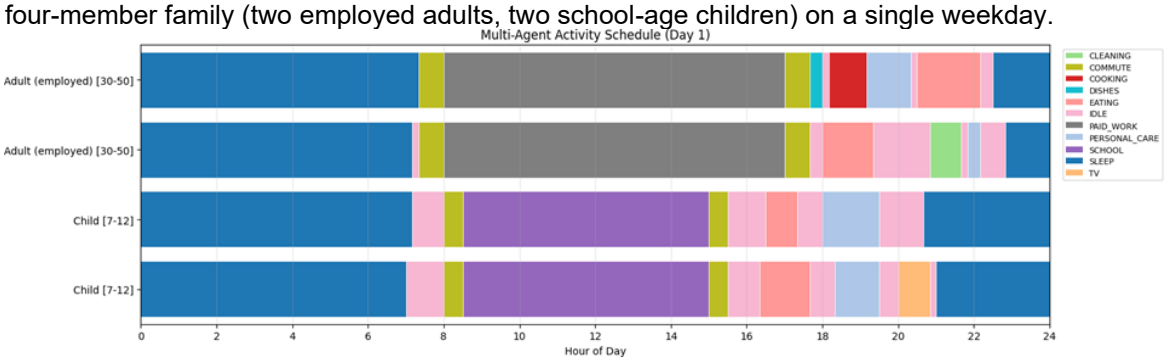

Figure 5: Constraint-based activity schedule for a four-member household on a weekday. Adults have work blocks (grey), children have school (purple), meal times are synchronised, and flexible activities (cleaning, TV) fill residual time.

Finally, scheduled activities are translated into electrical demand profiles. Appliance operation is modelled independently from human interaction duration, allowing appliances such as washing machines or dishwashers to continue consuming electricity after user interaction has ended. Additional components include always-on loads, occupancy-dependent lighting, domestic hot water demand, EV charging, and space-heating demand.
Space heating is calculated using the ISO 13790 5R1C thermal network model, which captures building thermal dynamics while maintaining computational efficiency. Heat-pump performance is represented using empirically fitted performance curves derived from certified heat-pump data. EV charging behaviour is based on probability distributions derived from large-scale Swedish charging-session datasets.

2.3.2 Calibration and Validation

The simulator was calibrated using multiple Swedish statistical and technical data sources. Household demographics were based on Statistics Sweden (SCB) household statistics, while activity participation rates and durations were derived from Swedish time-use surveys. Heating-system distributions and appliance characteristics were calibrated using data from the Swedish Energy Agency and other Swedish building-energy references.

A distinction was maintained between calibration and validation. Calibration involved adjusting model parameters to reproduce known benchmark values, while validation was conducted using independent datasets and external reference models.

The appliance-level electricity demand was validated against Swedish Energy Agency benchmark values for seven major appliance categories. After calibration, all categories were reproduced within ±25% of target values. Figure 6 presents a visual comparison of the final (post-calibration) simulated versus target loads by category.

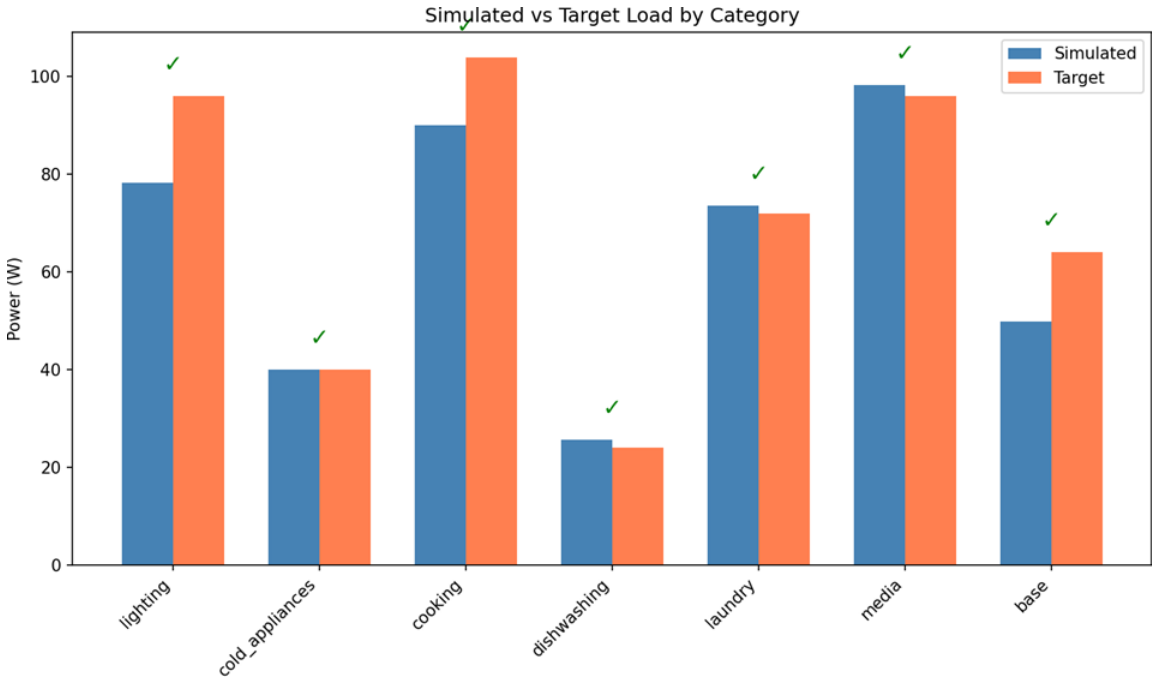

Figure 6: Simulated versus target average power by appliance category (10-run Monte Carlo average). Green ticks indicate all seven categories fall within 25% tolerance.

The thermal model was validated against Swedish TABULA reference buildings, achieving strong agreement with published benchmark values with a mean absolute percentage error of approximately 6.8%. Additional cross-validation was performed against EnergyPlus simulations. Initial discrepancies related mainly to ground-temperature assumptions and internal heat-gain modelling, but after correction the agreement improved significantly.

Finally, the simulator was compared against real anonymised smart-meter data from Swedish households. The results showed good agreement in both load magnitude and temporal structure. In particular, the simulator reproduced characteristic Swedish residential load patterns, including morning and evening peaks, weekday/weekend differences, and seasonal variations. Figure 7 presents the seasonal validation for household HH-C1 (a family of four in a 98.5 m² apartment).

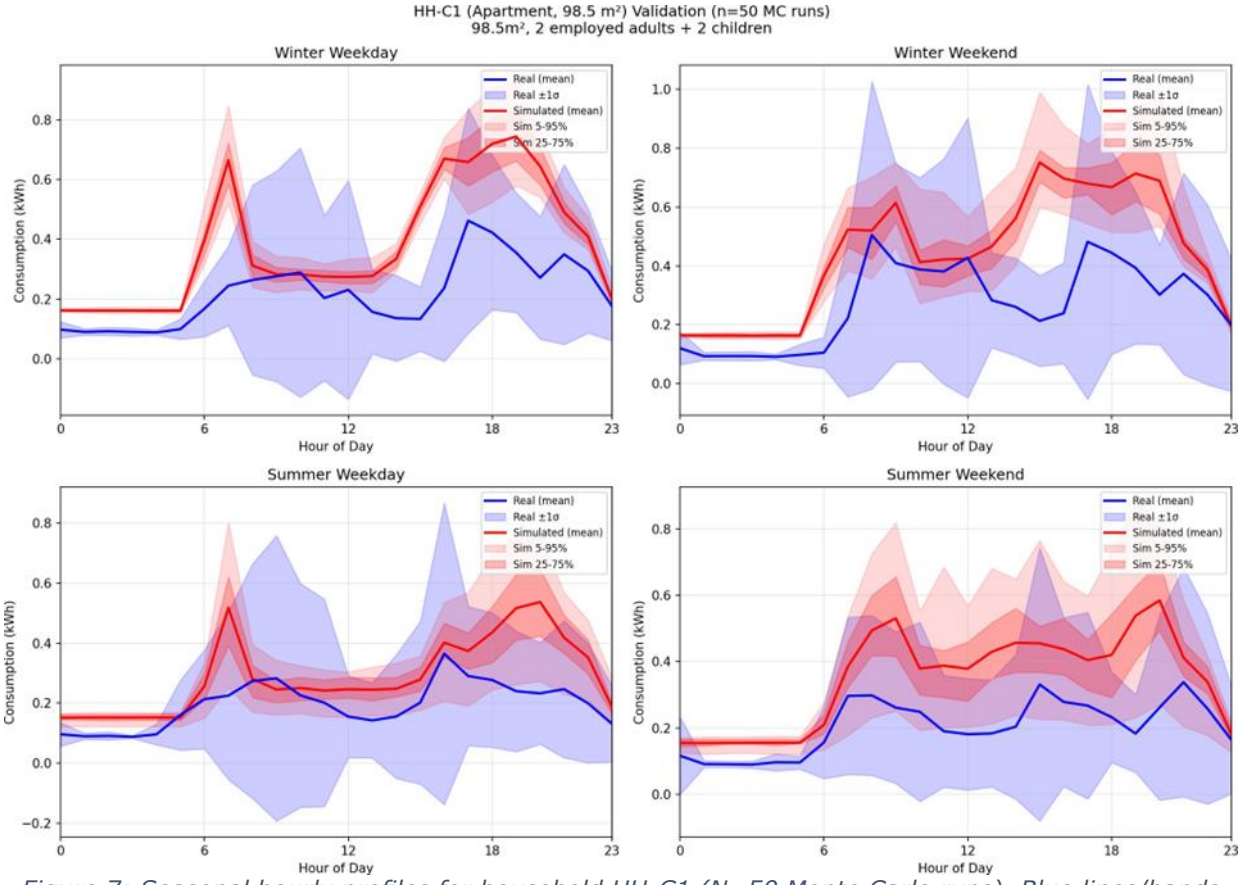

Figure 7: Seasonal hourly profiles for household HH-C1 (N=50 Monte Carlo runs). Blue lines/bands show measured data (mean 1σ); red lines/bands show simulated data (5th–95th and 25th–75th percentile). The simulated confidence bands overlap with measured data across all four season/day-type combinations. The characteristic evening peak (17:00–20:00) is captured in both timing and magnitude.

2.3.3 Results

The simulations demonstrate that the activity-driven modelling framework can generate realistic household electricity demand profiles at both individual and aggregated levels. Individual households exhibit characteristic "double-peaked" daily load patterns with morning and evening peaks, while weekend profiles show later morning activity and increased daytime occupancy.

Monte Carlo simulations revealed realistic day-to-day variability in both peak demand and total daily energy use. The stochastic framework also generated meaningful diversity effects at population level, which is particularly important for grid planning studies.

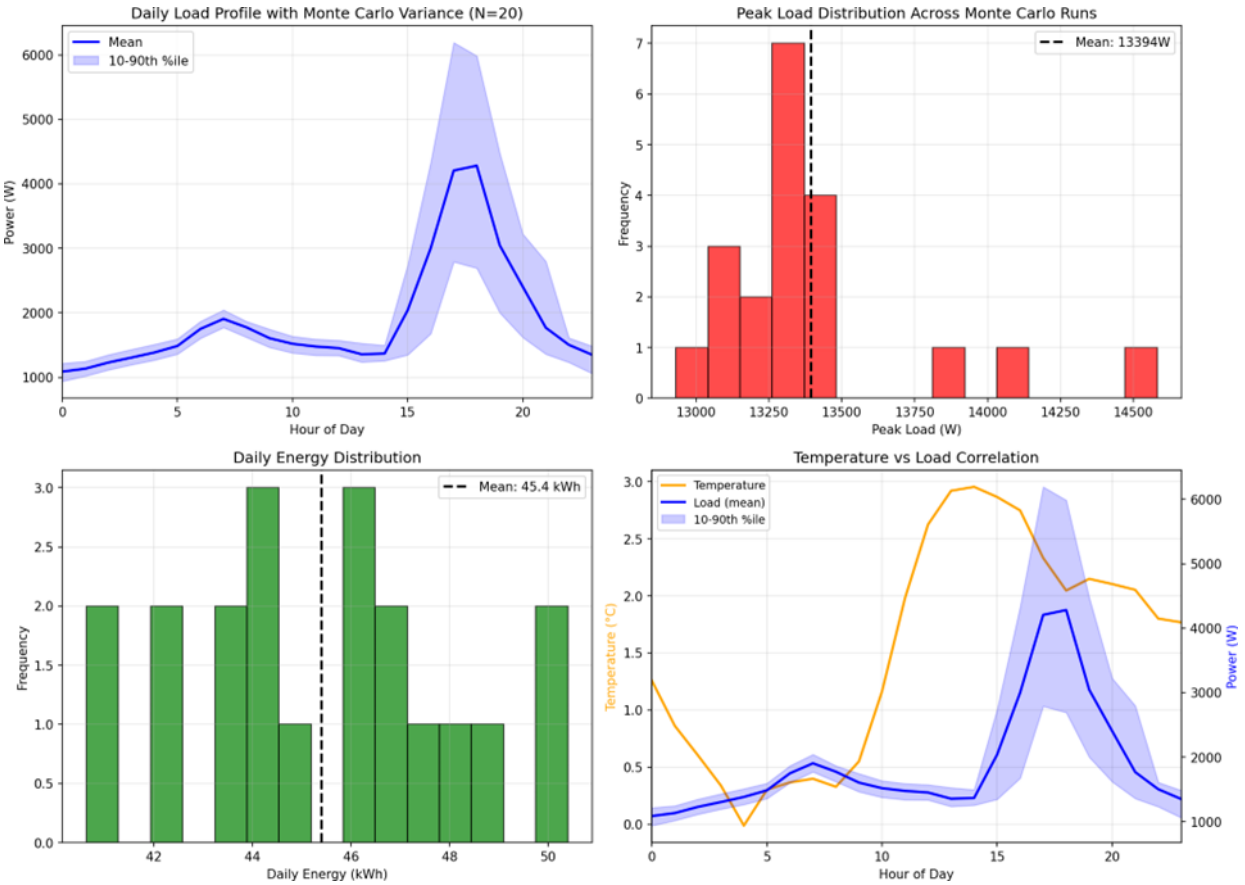

Figure 8: Monte Carlo variance analysis (N=20 runs) for a 140 m² villa with heat pump, two employed adults, two children, and an EV. Top-left: daily load profile with 10th–90th percentile band. Top-right: peak load distribution (13.0–14.5 kW range). Bottom-left: daily energy distribution (mean 45.4 kWh). Bottom-right: temperature vs. load relationship showing that the evening peak is behaviourally driven (cooking, EV), not thermal.

One important result concerns EV charging behaviour. Although increasing EV penetration substantially increases aggregate peak demand, the diversity factor remained relatively stable across different EV adoption levels. The stochastic spread in charging start times prevented complete synchronisation of charging loads. Nevertheless, the after-diversity maximum demand increased significantly as EV penetration increased, highlighting the importance of controlled charging strategies in future low-voltage networks.

Comparisons between heating systems showed clear differences in both energy use and electrical peak demand. District heating produced the lowest electrical demand, while heat-pump-dominated systems exhibited higher electrical peaks during cold periods. Simulations also demonstrated that behavioural loads such as cooking and EV charging contribute significantly to evening peak demand, sometimes more than space heating itself.

The results indicate that smart charging strategies for EVs may provide larger reductions in residential peak demand than improvements in building thermal efficiency alone. The simulations also confirmed the importance of diversity effects for transformer sizing, as aggregated household peaks were substantially lower than the sum of individual maximum demands.

2.3.4 Conclusion

SweLoadSim demonstrates that a bottom-up, activity-driven approach can successfully generate realistic synthetic household electricity load profiles for Swedish conditions. By

combining stochastic activity generation, constraint-based scheduling, building thermal modelling, and appliance-level electricity demand, the simulator captures both individual behavioural realism and population-level statistical consistency.

The validation results indicate acceptable agreement with Swedish benchmark datasets, thermal reference models, and real household smart-meter data. The framework is particularly well suited for scenario analysis involving electrification, EV charging, flexibility studies, and future energy-system planning where measured data is unavailable or insufficient.

Several limitations remain. Some appliance data are based on older reference studies and should be updated to reflect modern appliance efficiency. Temporal activity distributions are partly derived from pre-pandemic datasets and may not fully represent current work-from-home patterns. In addition, several behavioural dependencies between activities are not yet explicitly modelled.

Despite these limitations, the simulator provides a flexible and transparent research platform for synthetic residential load generation. Future work will focus on improved out-of-sample validation, integration of photovoltaic and battery systems, smart EV charging strategies, and enhanced behavioural modelling.

3 High time resolution load monitoring (UPB)

3.1 GreenMogo trail site description

GreenMogo is a zero-energy building operating in different scenarios to teach various target groups about more sustainable ways of living and building. Within this paper, the premise acts as a regulatory restricted prosumer (following the operation of a UniRCon) providing insights on energy needs for modern local energy communities. The general grid topology of this UniRCon is presented in the Figure below and consists of a family residential building (with regular home appliances) with PV generation and BESS interfaced with a Victron inverter and a heat pump for thermal energy transfer. The measurement layer consists of high-reporting rate information (1 frame/second), which is provided by energy meters operating using the Unbundled Smart Meter (USM) concept for energy use, and also provided by the Color Control GX TCP/IP integrated with the Victron Inverter for energy generation (PV and BESS).

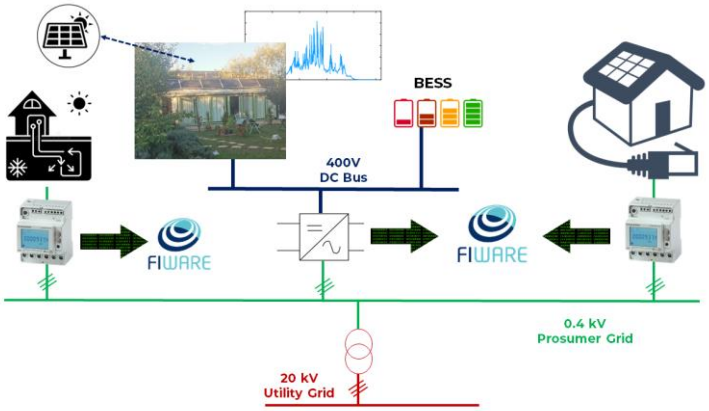

Figure 9: GreenMogo Framework

3.2 High-reporting rate information for loading

The information for the load side integrated in the FIWARE platform is derived from local measurements using the concept of the Unbundled Smart Meter (USM) set on 1

frame/second reporting rate. The classic architecture for the USM is in place consisting of the Smart Metrology Meter (SMM) part (metrology-compliant) and the Smart Meter eXtension (SMX) implemented using a low-cost processing unit. The GreenMogo UniRCon smart metering infrastructure streaming information for the FIWARE platform consists of three-phase meters installed on the premises. SOCOMEC three-phase meters, representing the metrology part of the USM, are used to monitor the energy transfer in the UniRCon environment, having a metering accuracy Class 1 for active energy and Class 2 for reactive energy. The SMX part, based on Raspberry PI (RPI) 3 board (acting as a single board computer) is physically connected to SOCOMEC meter allowing the extraction of high-time resolution instrumentation values (1s resolution). To achieve the high-reporting rate of 1 frame/second, the connection between the SMM and SMX is based on a serial interface (RS485), using the IEC 62056 (DLMS/COSEM) communication protocol, 9600 baud communication speed and additionally, for data extraction, specific configuration actions were required. One of the SOCOMEC meters is monitoring a residential power profile (consisting of standard home appliances) and another meter is installed as to observe the energy transfer of a three-phase heat pump (with nominal power 3 kW).

Examples for such power profile for the load are given in the Figures below.

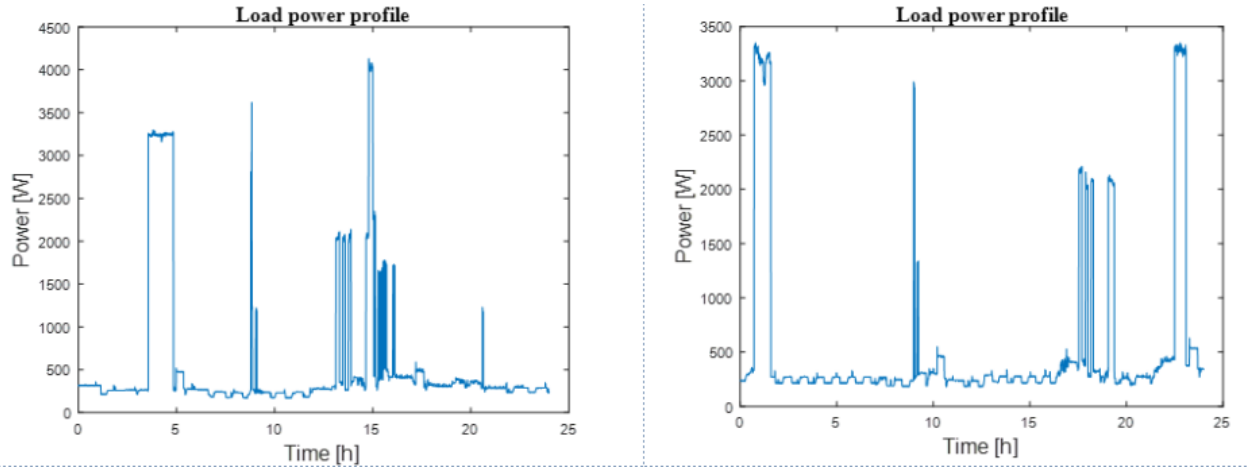

Figure 10: Power Profiles

3.3 High-reporting rate information for generation

The UniRCon environment is supplemented with PV panels generation, with 5.5 kW installed power and a local MPPT device to continuously maximize the output power of the PV panels. The flexibility in operation is provided by an energy storage system based on 3 lithium iron phosphate batteries with a built-in battery management system (BMS) summing up to 10.5 kWh. All generation equipment are integrated in a 5 kW Victron Quattro Inverter with Color Control GX (CCGX) control unit of the renewable generation. High-time resolution measurement information (1s resolution) is achieved by connecting a Raspberry PI (RPI) via TCP/IP protocol to the CCGX and performing dedicated software configuration based on Python with additional software calibration. For the acquisition of high reporting rate data from the local RES installation, additional software is needed since the manufacturers' software usually does not supply such high reporting rate data. On the Raspberry Pi, a Python script is running to collect the data locally in .csv files and forward the readings to the FIWARE platform. Generation information is available with a high-reporting rate of 1 frame/second creating the premises for heterogeneous data integration from multiple sources within the powered-by-FIWARE platform under development.

The Python script receives a reading roughly every 500 ms and averages over a second before storing and forwarding the data. The storing is done locally in a CSV file including the timestamp and basic load and generation parameters provided by the CCGX. After storing

the data, the same data point is forwarded to the FIWARE platform and can be used either for visualization or other third-party applications integration.

Examples for such power profile for the load are given in the Figures below.

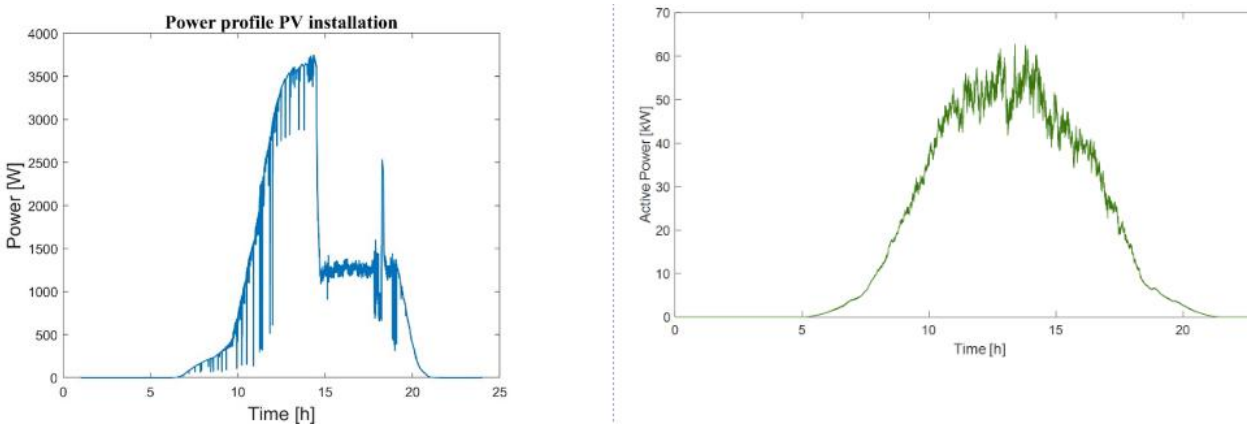

Figure 11: Power Profiles

3.4 Statistical approach for power profile assessment - using CV(RMSD)

The measurement process generates information by aligning the observed signal with its assumed implicit model. Recognizing the challenges in accurately assessing, reporting, and using information from low inertia power systems, especially when dealing with highly variable quantities during both the measurement and the reporting windows, we address in this work the overall uncertainty associated with classical measurements in power systems. The focus is on quantifying the information uncertainty derived from the variability of parameters describing the energy transfer which are usually silently assumed constant during the measurements reporting window. To this, we use statistical measures and propose a monitoring approach able to reveal – for the considered observation time, depending on the application - the particularity of energy transfer in a selected node/branch of power network, directly impacting the choice of measurement system. For the chosen application of the metrics below, the meaning of the variables is as follows: y_i– sample corresponding to the assumed model y, x_i– measured value; n – number of elements in the measurement series, $\overline{y}$ – an information concentrator describing the assumed model y during the selected process time window; for example, we choose the mean value of the measurand (x_i) on reporting time $T_r = 1/RR$ where RR is the reporting rate; $\overline{y}$ – an information concentrator describing the assumed model y during the selected analysis time window; we chose the mean value of the measurand on the analysis time interval $(T_a > T_r)$. To be noted that those metrics cannot be applied to "cold" processes where the measurand is not existing (for example, zero power transfer).

The coefficient of variation (CV) of the Root Mean Square Deviation (RMSD) is a statistical measure that provides a normalized measure of the variability of RMSD values. The RMSD is often used in the context of assessing the difference between predicted (y_i) and measured values (x_i) [5].

$$CV(RMSD) = \frac{1}{\overline{y}_p}\sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}$$

where $\overline{y}_p$ refers to an assumed model value representative for the selected process time window.

CV(RMSD) is used to express the relative variability of the root mean square error. It is important to note that, in our case, they are similar metrics because the assumed model

value $\overline{y}_p$ is also the mean $\overline{y}$ of the model samples y_i calculated on T_r; the choice of representative model value $\overline{y}_p$ depends on the context of application [5], for example it can be chosen the rated power of the grid connection.

For the assessments presented in this paper, four consecutive days during a week were selected (including weekend) the metric CV(RMSD) was applied for three reporting rates, corresponding to T_r of 15 min, 30 min and 1h, on several daily power profiles recorded in July. One of the power profile is depicted below.

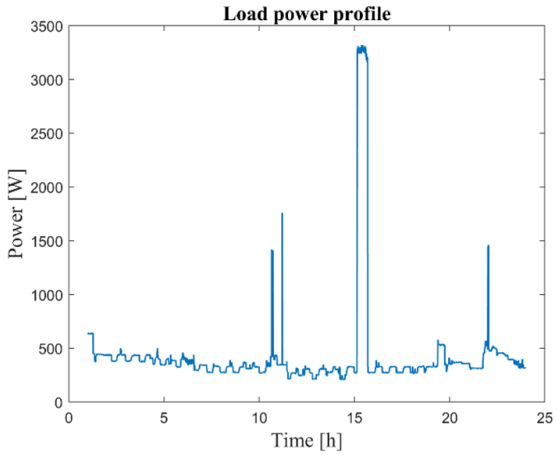

Figure 12: Power Profile

Figure below presents the metric CV(RMSD) applied to load power profiles for a daily analysis window (24 hours) and hourly constant profile model (Ta = 1 hour), of the profile above. One can observe that the maximum CV(RMSD) is 122% depicted at 4:00 pm, which corresponds to the 16th value, on Tr16. In addition, the Figure below presents the metric CV(RMSD) applied to load power profiles for an analysis window Ta = 24 hours and Tr = 30 minutes. One can observe that the maximum CVRMSD is 130% depicted at 3:00 pm, corresponding to *Tr30*. More over, the third picture below presents the metric CV(RMSD) applied to load power profiles for an analysis window *Ta* = 24 hours and *Tr* = 15 minutes. One can observe that the maximum value is 121 % depicted at 3:30 pm, corresponding to *Tr62*

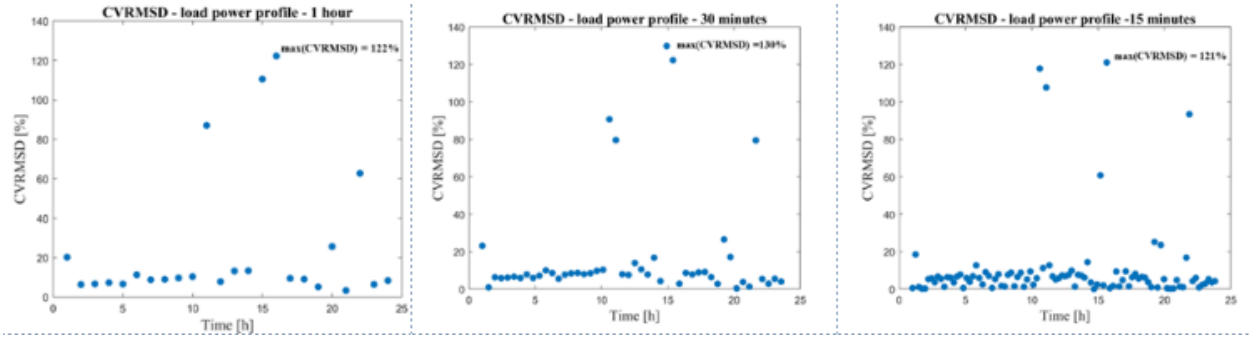

Figure 13: CV (RMSD) Analysis

3.5 Estimation of the LV power profiles variability using the Goodness of Fit approach

For this part of the study, we assess the active power profiles variability in the prosumer setup with $P_{max} = 8kw$ during a spring month in 2023 for the two selected time reporting windows. We used GoF as defined in (5). One-day load power profile 01.03.2023 is presented in Figure below.

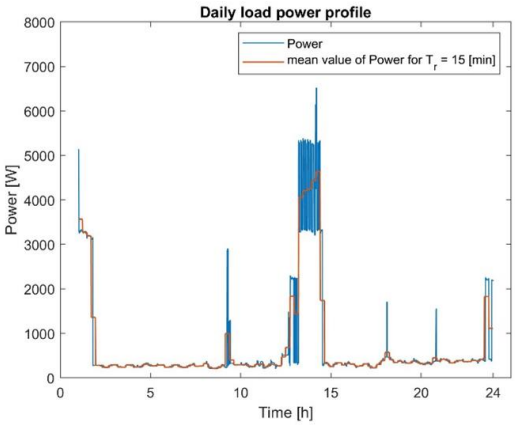

Figure 14: Power Profile

The GoF results obtain for the power profile above are illustrated in Figure below, using Tr=2 h. It can be observed that the maximum value of GoF is 12.29 dB at 14:00.

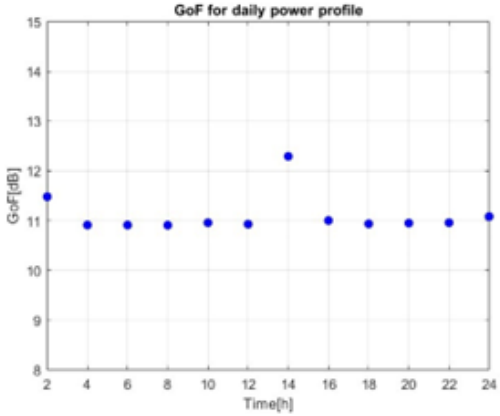

Figure 15: GoF Analysis

We repeat the procedure for one month (March 2023) to calculate daily GoF results using an analysis window Tr = 2 h (n=7200) and present them in Table below. One can observe that the maximum value is 12.61 dB on 24/3/2023. The minimum value is 10.83 dB for 26/03/2023. The median value is around 9 dB for the analysed month. In addition, we can estimate daily GoF with one single value using:

$$GoF^* = 20lg\frac{P_{max}}{|\bar{P}_i - P_{max}|}$$

where $\bar{P}_i$ is the daily average of the recorded load profile

Day	GoF [dB]			GoF* [dB] daily value
	min	median	max	
1/3/2023	10.86	11.06	12.26	1.75
2/3/2023	10.93	11.26	12.54	0.42
3/3/2023	10.82	11.18	11.90	0.33
4/3/2023	10.84	11.10	11.92	0.12
...	...	...	...	...
24/3/2023	10.84	11.38	**12.61**	1.55
25/3/2023	10.84	11.11	11.89	**2.41**
26/3/2023	**10.83**	11.01	11.87	**0.09**
...	...	...	...	...
30/3/2023	10.84	11.10	12.08	0.12
31/3/2023	10.84	11.05	11.98	0.30

One can observe in Table above that the GoF* maximum value is 2.41 dB on 25/03/2023 and the minimum value is 0.09 dB in 26/03/2023. Fig. above presents the daily ⟦GoF* results obtained for one month in March 2023. One can observe that most of the days have around 1 dB and we can identify only one day with ⟦GoF* above 2 dB. For a better understanding, we calculate also daily GoF for one day 01.03.2023 using 4 frames/h, which is (still) the legacy sampling rate for smart metering today. The comparing results for both 1 frame/s (Nr=7200) and 4 frames/h (Nr=8) are illustrated in the Figure below. Small differences can be observed between the results, this indicates that the legacy sampling rate used in smart metering (4 frame/h) is adequate for assessing power flow variability.

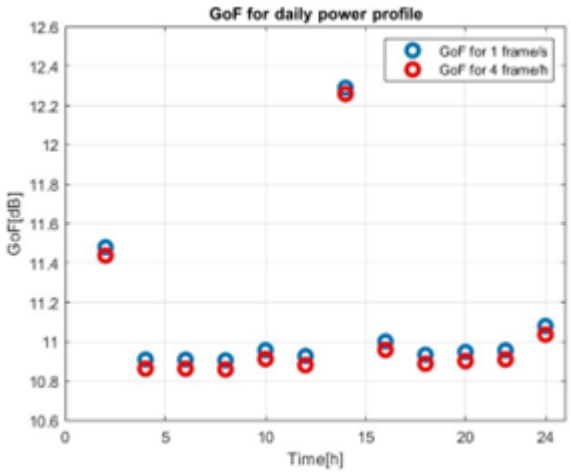

Figure 16: GoF Analysis

3.6 Leveraging Anomaly Detection and AutoML for Modelling Residential Measurement Power Traces

By combining anomaly detection with automated machine learning, in the context of power systems measurements, we aim to improve the data analytics approach for such data and contribute to the real-time operation and control of local microgrids. The presented use case is focused on single and multiple residential units (student dormitories) by using a previously deployed open metering infrastructure that enables data collection and aggregation.

A generic definition of anomalies in power system measurement traces can be seen as "patterns that deviate from a well defined notion of normal behaviour". In order to partially

mitigate the challenge of accurately identifying and labelling a large number of anomalies in energy and power measurements time series, the authors proposed a method of modelling and generating synthetic anomalies to improve machine learning model training. Such approach can be validated through the statistical properties of the anomalies being considered or through the help of domain experts. Bootstrapping or semi-supervised methods are also available in which a small number of labelled anomalies are available and their characteristics are used to extrapolate and identify a larger number of occurences. A high performance method for time series data analytics (TSAD) based on the Matrix Profile data mining algorithm is presented. The application concerns online streaming data for which real time results are important through Discord Aware Matrix Profile (DAMP) while the authors argue an analytics throughput of over 3000000Hz on standard hardware.
Automated machine learning (AutoML) pipelines for fast and practical evaluation of classification and prediction performance is being increasingly used in diverse engineering areas. Given a well formulated problem and an initial set of models and hyperparameters, the approach can guide the user to a suitable working solution. A reference is provided in where building energy forecasting is carried out with good quantitative performance. The system is complemented by explainable artificial intelligence (xAI) features that can improve the understanding of the outputs by the end-user, identifying the most important features in the determination of the final result. Such methods can integrare both baseline and state-of-the-art algorithms and combine their outputs, e.g. through ensemble voting schemes, in order to provide robust forecasting outputs over a wider range of input variability.

A diagram of the proposed system is shown in the Figure below

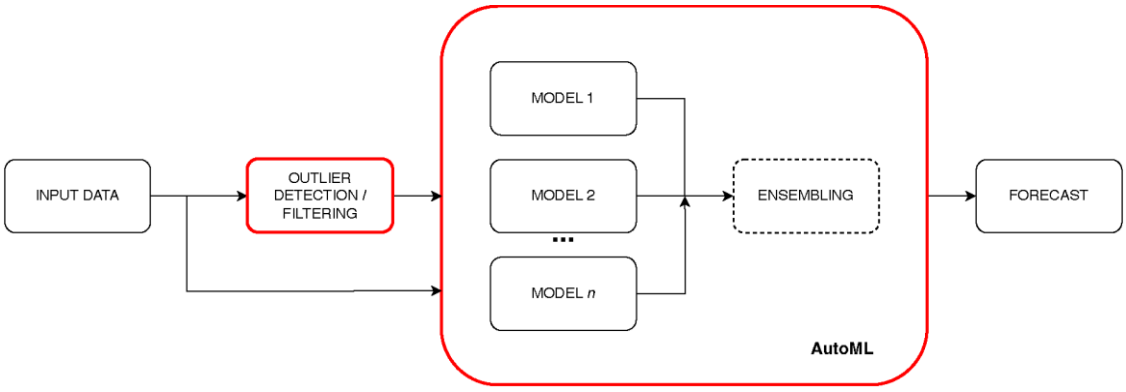

Figure 17: Anomaly Detection Process Flow

The main four stages in the data processing pipeline are described:

- Input Data: Power measurement values are read in offline mode from text files containing the readings and associated timestamps or directly, in online mode, by querying the metering infrastructure or through database Application Programming Interfaces (API); the values are stored in structured format e.g. dataframe format, in the development environment for the next stage;
- Outlier Detection / Filtering: The anomaly detection and labelling routing is run on the structured and cleaned data; The datapoints that are identified as outliers are properly marked
- AutoML: The automated machine learning procedure involves training a number n of distinct parametrised forecasting models on the datasets that include and exclude the outliers from the previous steps; As an optional sub-step, the predictions of the different models can be combined by computing a weighted average of the model predictions, using the inverse of the Mean Squared Error (MSE) metric as weight; This step allows the quantification of the outlier removal on the forecast accuracy;
- Forecast: The final forecast result is presented to the end-user or provided to the decision and control system for further processing.

Active power measurement datasets are used to illustrate the approach as multiple housing unit building (student dormitories) from the campus of the Politehnica Bucharest, Romania. Four days of active power measurement from the dormitories with a 2s resolution are illustrated in Figure below. Large seasonal variations are also observed based on the heating/cooling requirements in winter/summer compared to shoulder seasons.

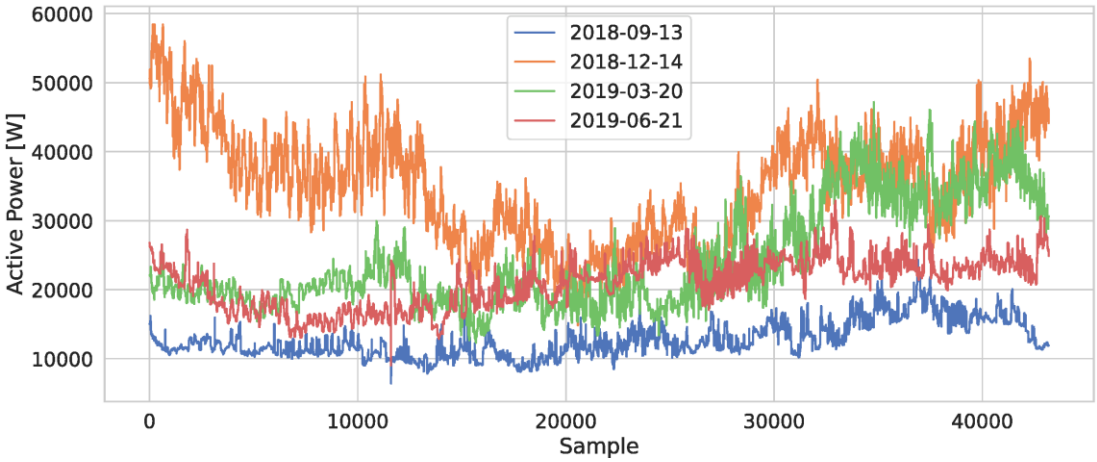

Figure 18: Politechnica Bucharest, Romania Campus Trends Overview

For obtaining the results, implementation has been performed in Python in the Google Colab hosted notebook environment. The data and the Jupyter Notebook code is available on GitHub for replication of the Figures and result tables. Main steps included data import and pre-processing, e.g. timestamp formatting, outlier filtering and automated time series forecasting. An initial example for using Hampel filtering for outlier detection on the dormitory data from September 13th 2018 is introduced in the Figure below. The red points identify the data points in the original timeseries for Figure 14 that have been labeled as outlier using the current configuration of the Hampel filter. The remaining blue line depicts the timeseries with these outliers removed.
The overall outlier rate by using the standard parameters: threshold for number of standard deviations (3) and neighbor window size (15), ranges between 5 and 6 % for the analysed days.

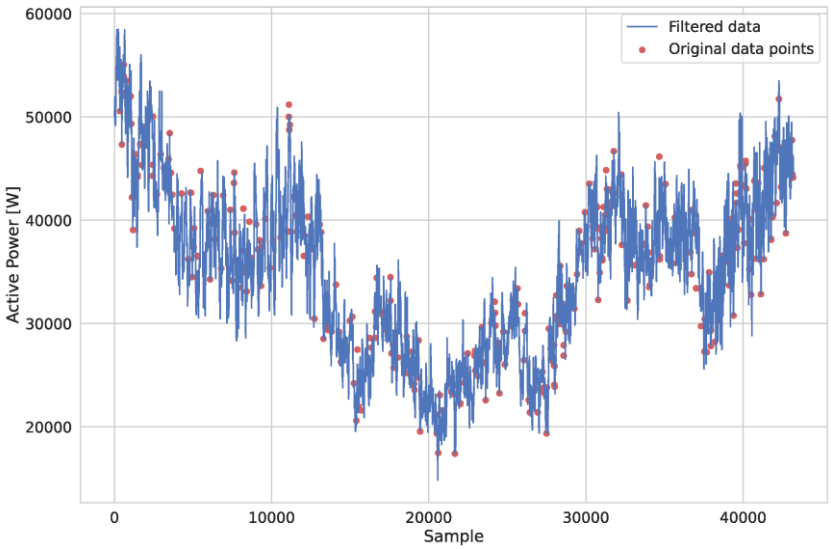

Figure 19: Hampel Filter Analysis for Outlier Detection

We test the auto time series modelling method on both the original and the Hampel filtered data with the following configuration for the training stage:

- model = AutoTS(

- forecast_length=3,
- frequency='infer',
- model_list='probabilistic',
- ensemble=None,
- max_generations=3,
- num_validations=2)

where model_list denotes the subset of available models in the auto-ts library, with a number of 430 models for the probabilistic option. For computational efficiency, we do not use the ensemble option while the number of validations is set at 2, for improving model selection with limited penalty on the performance. The sampling rate of the input time series is inferred automatically from the DateTime index.

Figure below shows the day ahead forecast using the original and filtered data for training at 20s time steps. This allows the qualitative assessment of the prediction performance for the original and filtered - outliers removed, input data, while the quantitative metrics were based on Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage loss (sMAPE) and Scaled Pinball Loss (SPL), or Quantile Loss.

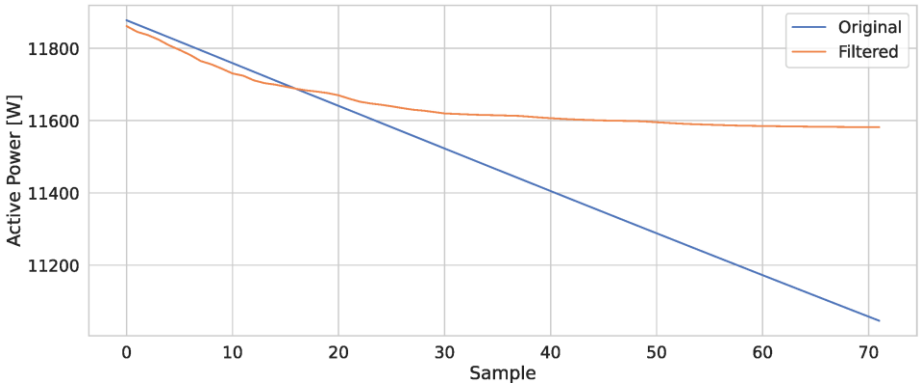

Figure 20: Prediction Outcomes

4 Data driven profiling and forecasting (DWH)

4.1 Austria Energy Community (EC1) Electricity Consumption Analysis, Forecasting and Optimization (DWH)

The Energy Community (EC) in one of the provinces in Austria is a small community of 5 different participants where two of them have a PV-System with included storage capacity. The other 3 are passive members of the EC without any flexible resources. In total we received the data of 1 year and 9 months for 4 of the participants. The fifth participant has joined later and is excluded from the analysis, since we do not have enough data to train models. Therefore we proceeded with 4 participants in total.

Based on the data gathered, dwh GmbH has performed basic data analysis, pattern recognition, identification of potential optimization opportunities, and forecasts of the EC participants' energy consumption and PV production. These are furthered to the optimization tasks in WP4. The total process is displayed in Figure 21.

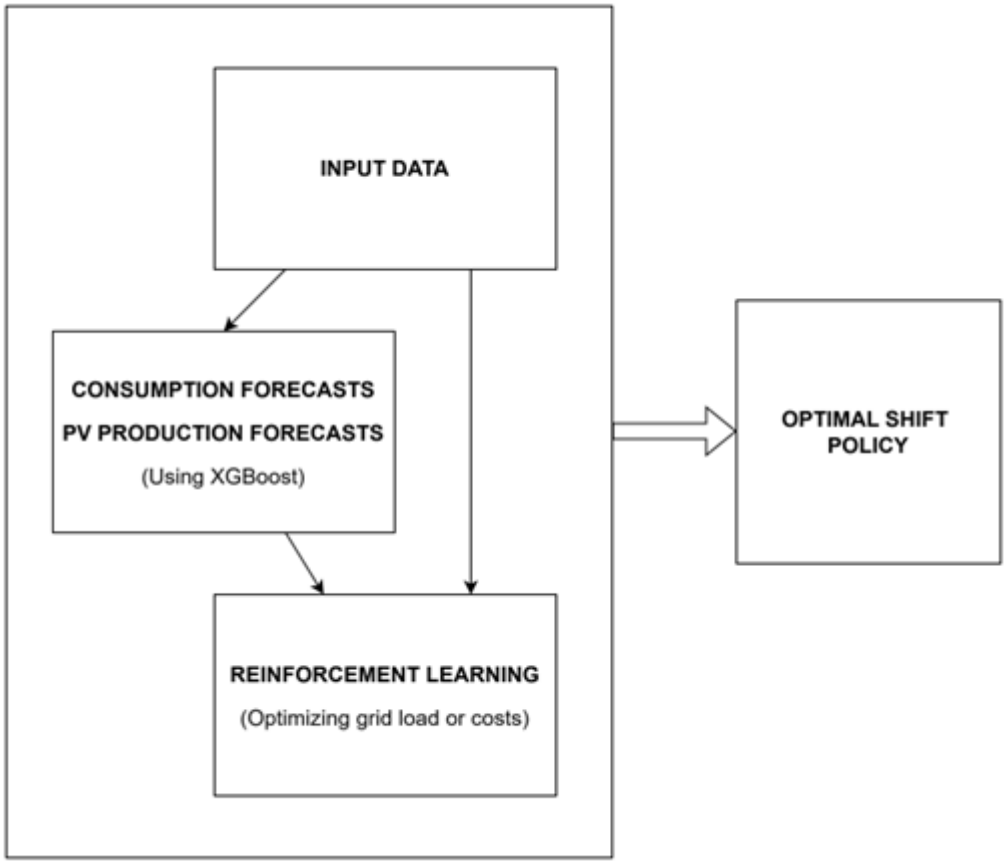

Figure 21: Overall Forecasting and Optimization Process for the EC

4.1.1 Data Analysis

The basic data analysis indicates unrealized potential for optimization in battery management. As can be seen in Figure 22, the battery was charged using grid energy several times, even though the state of charge did not approach low percentages of the battery.

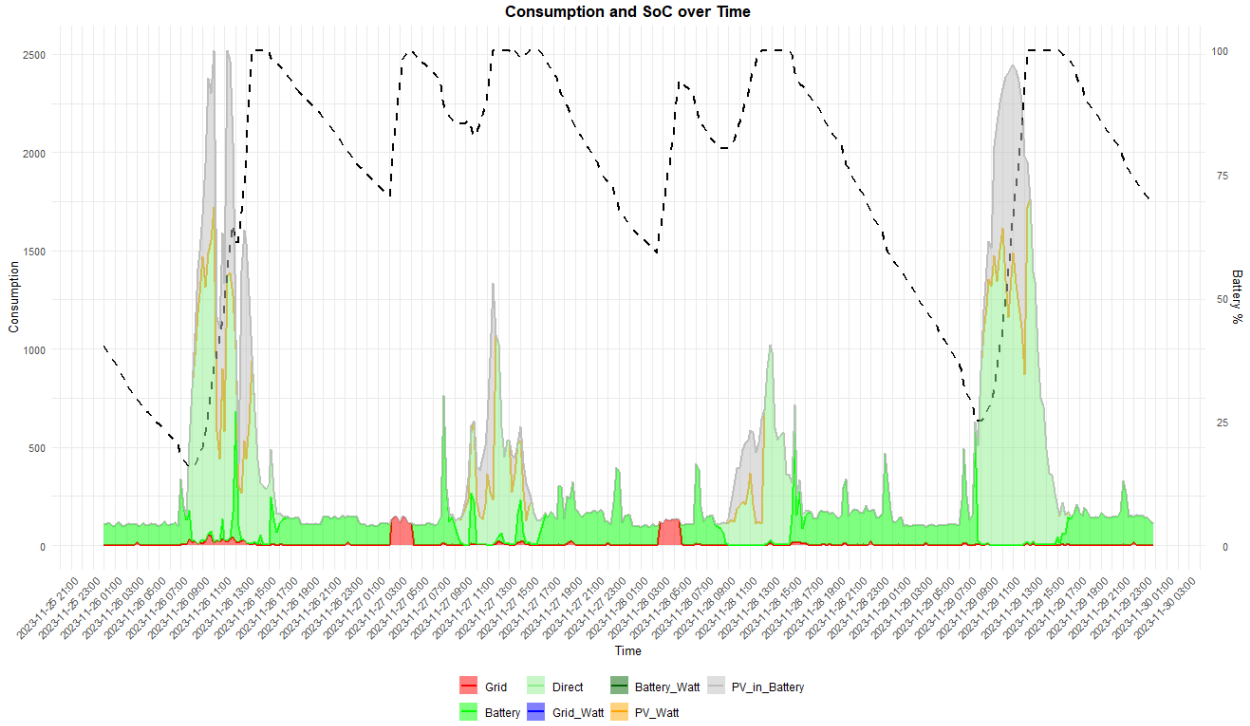

Figure 22: Optimization Potential for Participant without Wattpilot

Another participant (see Figure 23) employs a Wattpilot charging station in addition to a battery. In this case we can again identify potential for optimization. On the first shown day, the Wattpilot is charged using grid energy even though the battery's state of charge is at 100%. Later, we also observe that the battery is not fully emptied. On the last day, for example, the entire battery capacity is used for charging, after which grid energy is required throughout the night.

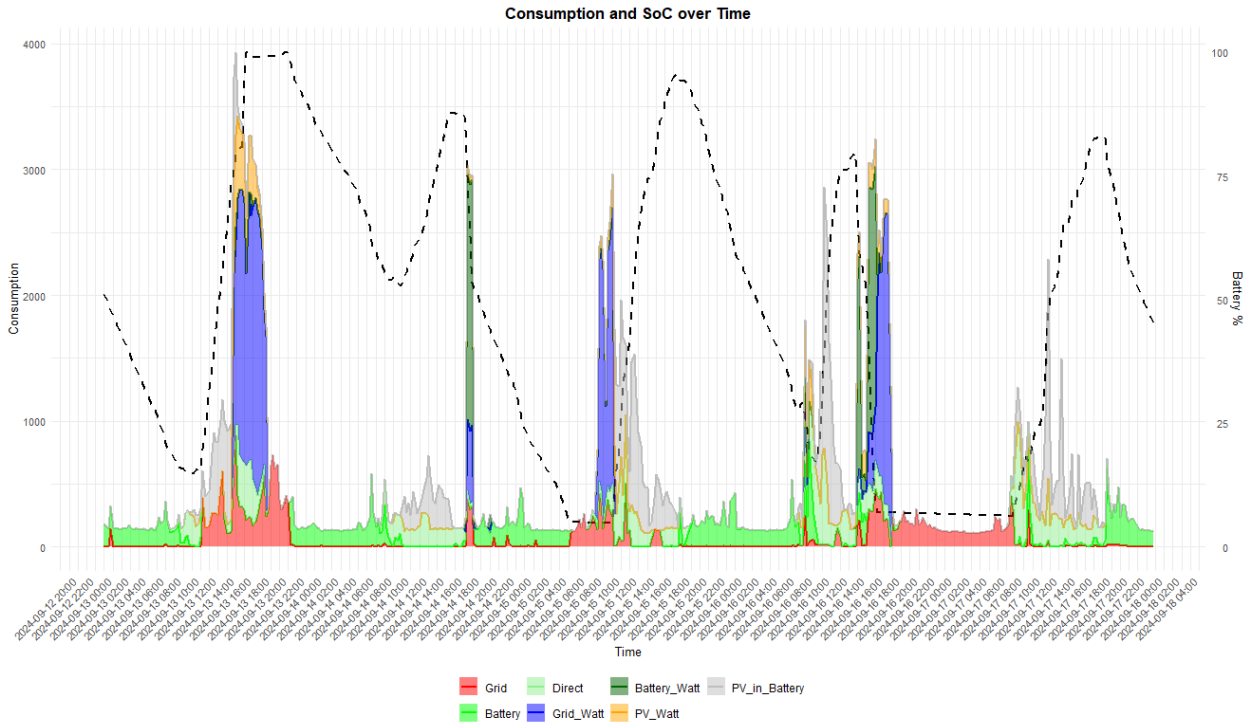

Figure 23: Optimization Potential for Participant with Wattpilot

A more advantageous strategy can be seen in the following example (see Figure 24), where the charging of the Wattpilot was stopped in order to maintain a sufficient state of charge for the night.

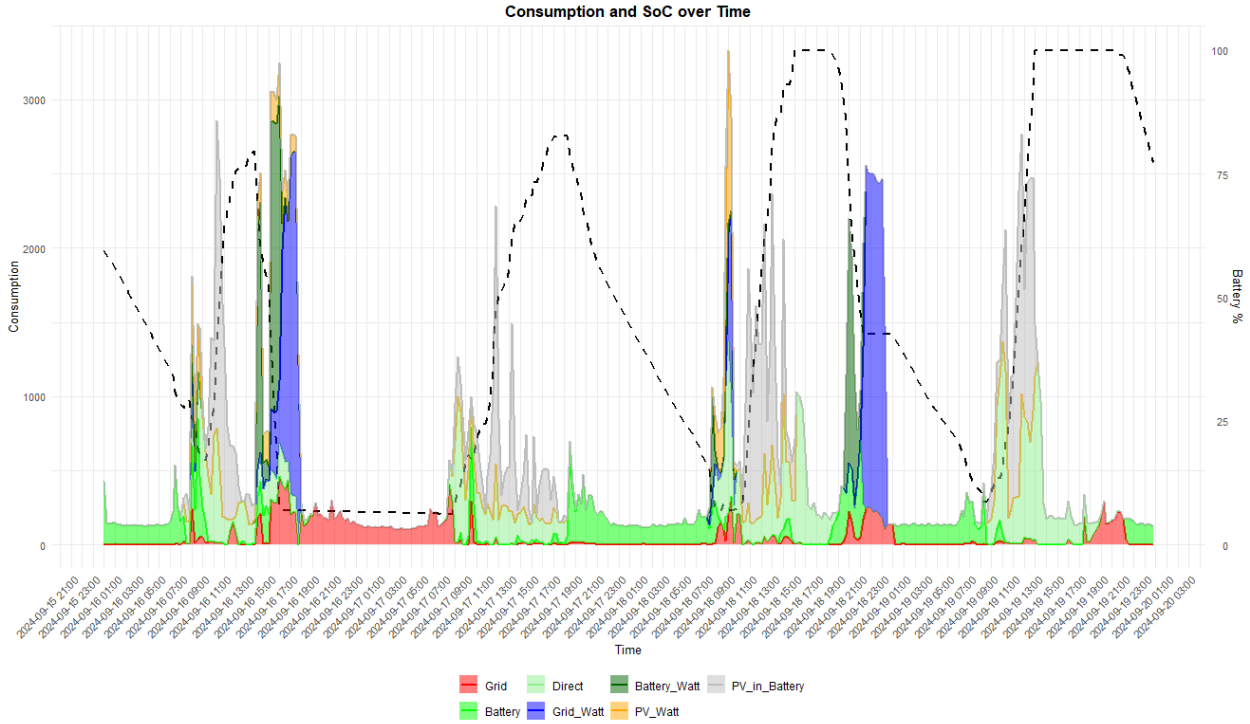

Figure 24: Potential Fix on how to Optimize the use of Battery and Wattpilot

A detailed analysis of the data reveals several such patterns, indicating that there is a lot of potential for optimization through load shifting.

4.1.2 Consumption and PV Production Forecasting

For the forecasting task, several data sources are integrated to improve prediction accuracy. This results in a variety of different explanatory variables:

Household-level variables:

- Grid import per household
- Grid export per household (if available for the participant)
- PV production per household (if available for the participant)

External / auxiliary variables:

- Weather data (solar irradiance, temperature)
- Time features (seasonal cycle via sine/cosine transformation)
- Grid prices for import and export (used for the optimization task later)

Time resolution and coverage:

Data was collected from 01.01.2023 to 01.10.2024, providing more than 1.5 years of data. This dataset was split into a training and test set, with the last 12 weeks reserved for testing and the remaining period used for training.

Depending on the source, different granularities are available:

- PV dataset from smart meters: 5-minute resolution

- EDA-report dataset: 15-minute resolution
- Weather data (Geosphere): available in 10-minute or hourly intervals

To combine the data sources, all datasets were resampled to a common 10-minute resolution using weighted linear interpolation for the EDA-report data and simple aggregation for the pv datasets. Further calculations are all done by using the newly created 10-minute resolution dataset

After preprocessing, a range of time series forecasting approaches was evaluated, including statistical models (ARIMA, SARIMA, SARIMAX) as well as machine-learning-based methods (XGBoost, Random Forest Regression, Support Vector Regression, K-Nearest Neighbors). Following a performance evaluation, XGBoost was selected as the most suitable modeling approach.

To ensure optimal model performance, a grid search was employed to identify the best hyperparameter configuration. In addition, two forecasting strategies were compared: (1) step-wise forecasting and (2) direct 24-hour forecasting, the latter being the primary objective of this work. Forecasts for both energy consumption (see for example Figure 25) and PV generation (if available for the participant) were produced and are used as input for the optimization tasks in WP4.

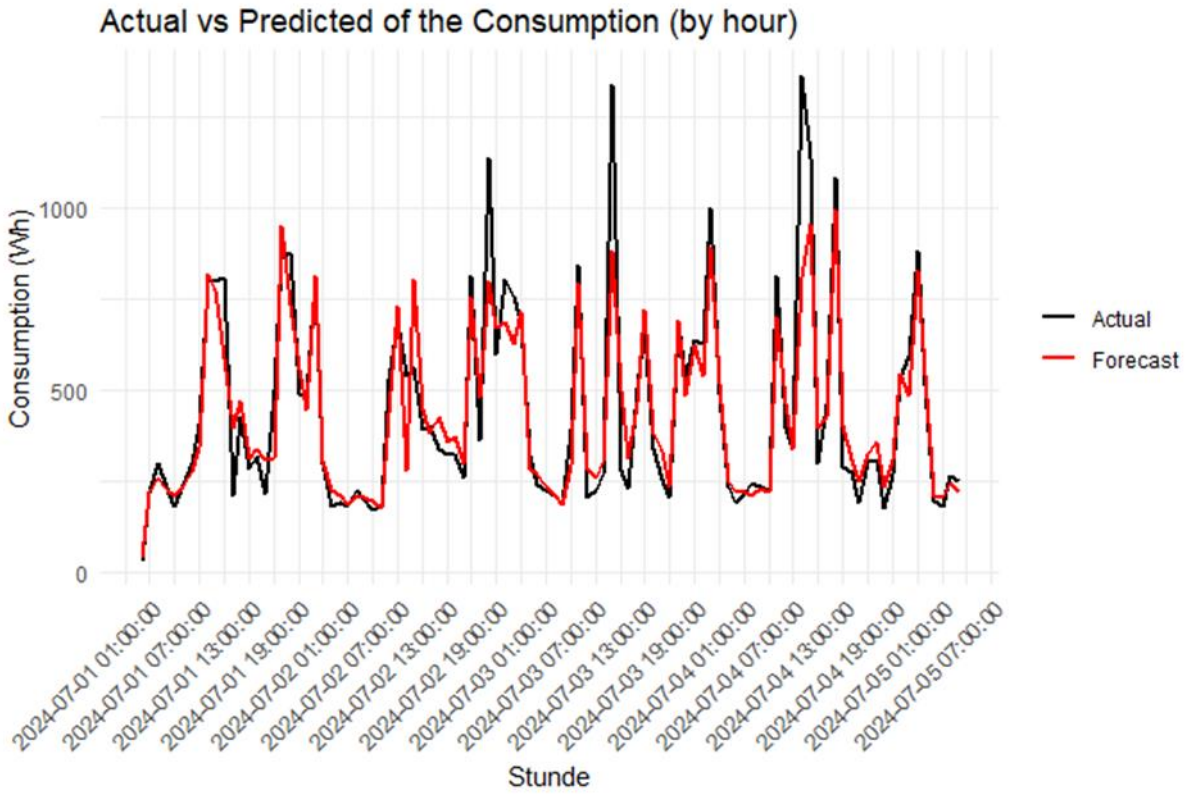

Figure 25: Consumption Forecasts Snipped of the Actual vs Predicted Plot

4.1.3 Reinforcement Learning Based Optimization (WP4)

Reinforcement Learning (RL) is applied to optimize energy consumption and cost reduction strategies within the energy community. While RL offers wide flexibility and the potential to discover non-obvious control policies that might not emerge from classical optimization methods, it requires large amounts of training data and long computation times. In addition, the resulting models can be difficult to interpret due to their black-box nature. The general structure of the RL process is highlighted in Figure Error: Reference source not found.

As an initial step, we focused on a single participant of the energy community. The first optimization objective was to minimize grid load. Subsequently, grid price information was introduced, and the objective was extended to minimizing energy costs for the participant. The policies learned by the RL agent were evaluated against both a rule-based baseline model and the actual operational behavior observed in the energy community. In a further step, the approach was extended to a Multi-Agent Reinforcement Learning (MARL) framework. This allows multiple participants to interact and coordinate their behavior, potentially improving overall system performance.

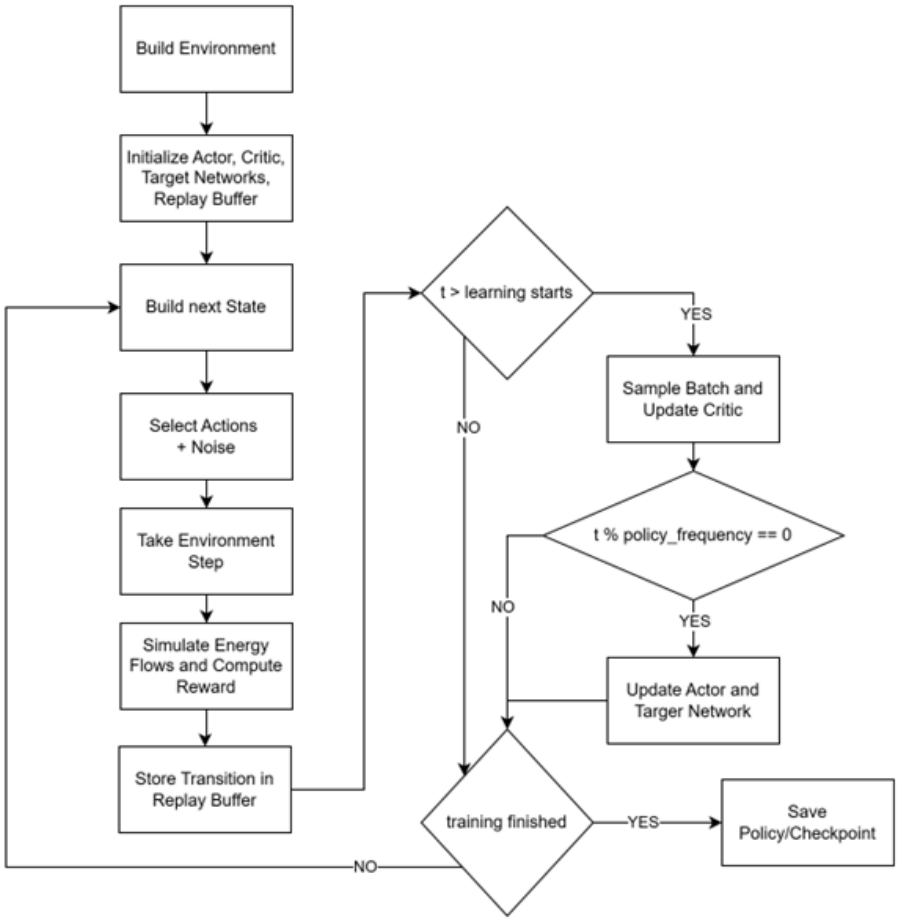

Figure 26: RL Flow Chart

4.2 Austria Energy Community (EC2) Electricity Consumption Analysis and Forecasting (TU Wien, DiLT, dwh)

The Renewable Energy Community (EEG) is centered around utilizing photovoltaic (PV) systems to generate solar power. Initially, the district's PV installations produced more electricity than local households consumed, sending the surplus back to the public grid. Thanks to Austria's Renewable Expansion Act (EAG) of 2021 and amendments to the Electricity Industry and Organization Act (ElWOG), this surplus electricity can now be shared across property boundaries with reduced network fees, making locally produced PV electricity cheaper than traditional public providers. Established as a formal association in July 2022 and fully organized by March 2023, the EEG includes diverse members such as private individuals, small businesses, and public institutions. It aims to expand by adding more PV installations, an electric vehicle charging station, and potentially an e-car sharing service, while fostering partnerships in the community through a grassroots approach.

Electricity consumption data for the energy community participants, covering approximately eight months, has been supplied to the consortium by one of our partners, DiLT Analytics. This data has a 15-minute time resolution. There are 19 participants in total, but we only have information on five (participants IDs: 4, 10, 13, 14 and 19) who have PV systems installed, although we currently lack access to their electricity generation profiles. In this analysis, we combine the community's overall electricity consumption with individual participants' usage profiles to identify key insights and explore the potential for forecasting electricity demand for each participant.

4.2.1 Energy Community Electricity Consumption Analysis

The data includes participants who lack available measurements, so they were excluded from the analysis because imputation is not feasible without additional predictor variables, such as non-operational factors (like demographics or routines) or operational data. The only operational data we have is electricity consumption for this community. For participants with missing measurements in certain months, we used Multivariate Imputation by Chained Equations (MICE) to estimate and fill in those missing values. Figure below illustrates the total energy consumption of a community over time from January to August 2024, showcasing a clear seasonal trend. In the initial months (January to February), energy consumption is high, peaking between 6 and 8 kWh, likely due to increased heating needs during winter. From February to March, there is a noticeable decline in consumption, suggesting reduced energy requirements as temperatures warm. By April, consumption stabilizes at a lower level, fluctuating between 1 and 3 kWh, reflecting more consistent energy use through the spring and summer months. Occasional spikes in May, June, and late August suggest temporary increases in energy demand, possibly due to specific events or external factors. Overall, the consumption profile of the community reveals a significant reduction in energy consumption from winter to summer, indicating effective seasonal adaptation or conservation efforts within the community.

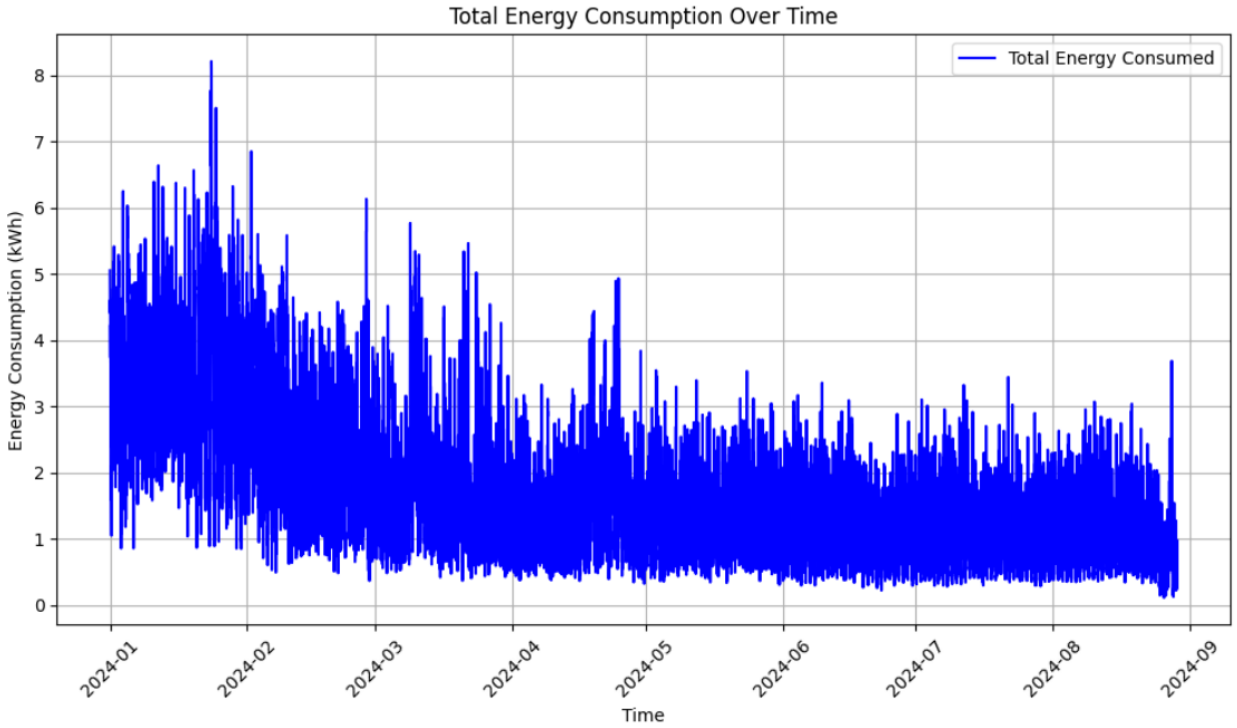

Figure 27: Total Energy Consumption (Aggregated 15-mins) of EC

In Figure below, the heatmap reveals distinct daily patterns in energy consumption across each hour. Generally, energy consumption is lower during the early morning hours (midnight to 6 hr.), indicated by lighter shades, suggesting reduced activity or usage during these times. As the day progresses, energy usage starts to increase, peaking around the evening hours, specifically between 18 hr. and 21 hr., where darker shades are observed. This indicates a trend of higher energy demand in the evenings, likely due to increased residential activity such as cooking, lighting, and other household routines. The mid-day hours, around 10 hr. to 13 hr., show moderate energy consumption, which may correspond to a mix of work and home activities but not as intense as the evening peak. The overall pattern shows a daily cycle of peak consumption during nights and lower consumption during sleeping and working hours.

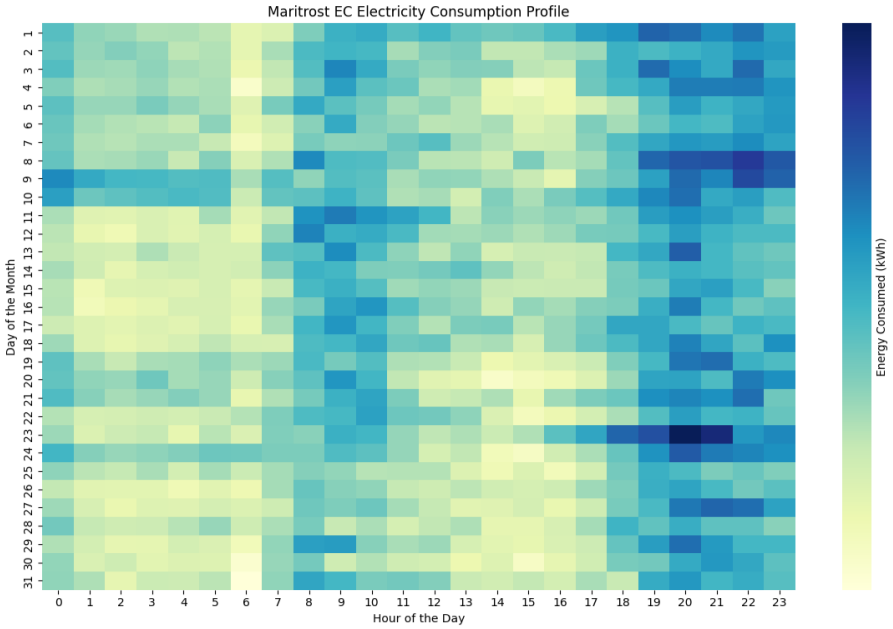

Figure 28: Electricity Consumption Patterns During Days of the Month

4.2.2 Participants Electricity Consumption Profile

Figure below presents peak energy consumption patterns among 19 participants. Participants 4, 10, 13, 14, and 19 have PV installations, which likely influence their energy use behavior. Despite having a PV system, Participant 10 shows one of the highest peaks at nearly 4 kWh, indicating substantial energy consumption, possibly during periods when solar energy is less available, like evenings, or due to significant high-energy events that exceed solar generation. Conversely, Participant 4, also with a PV system, has a notably low peak, under 1 kWh, suggesting effective solar energy utilization and minimal grid reliance. Participants 13, 14, and 19 exhibit moderate to low peak consumption, aligning with the expected benefits of PV systems reducing peak loads. This contrast, particularly between Participant 10 and others with PV, highlights differences in energy consumption habits, PV system sizes, or storage capacities. Overall, PV-equipped participants generally show reduced peaks compared to others, except for a few high-demand outliers, emphasizing the varied impact of solar power on energy behavior.

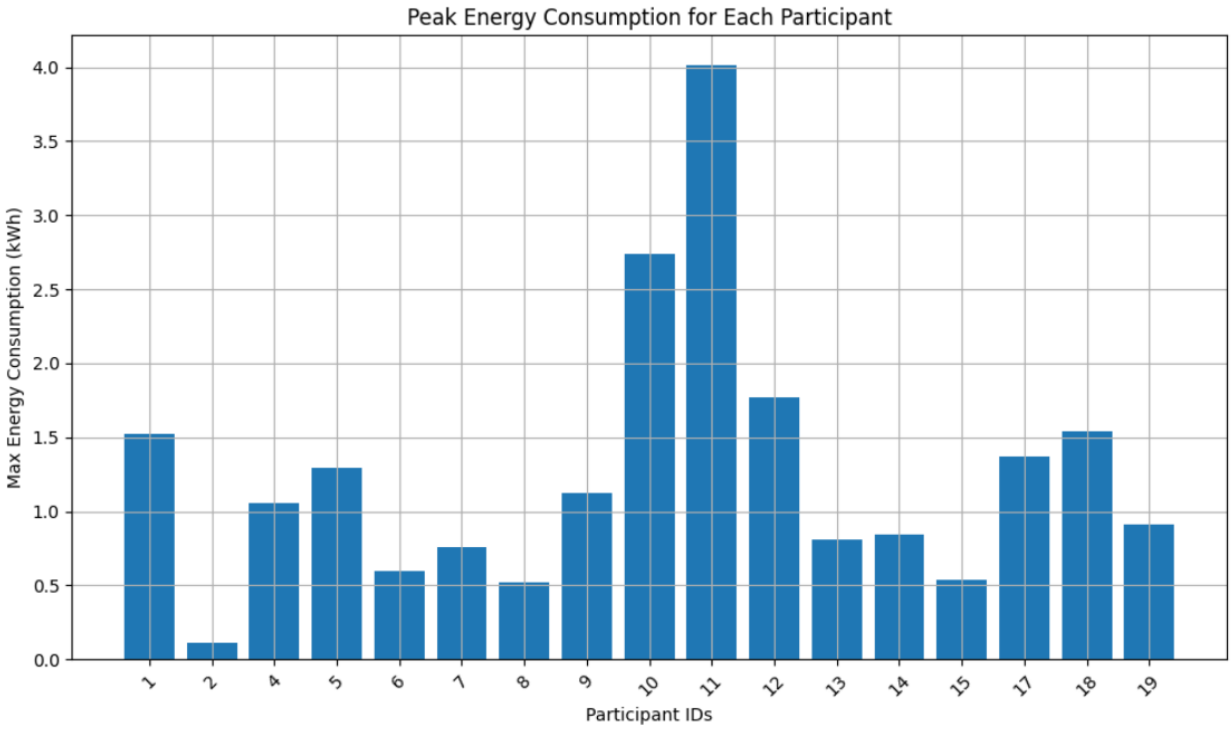

Figure 29: Max Energy Consumption for Each Participant

Figure below provides an overview for the electricity consumption of participants throughout the day in a month. The visualization indicates diverse energy consumption patterns among participants. Participants 1, 2, 4, 6, 7, 8, exhibit low and consistent energy use throughout the day, indicating a steady routine with minimal fluctuations. Others, such as Participants 10, 12, 14, 17, and 18 show high variability, suggesting a dynamic lifestyle with irregular schedules. Morning and afternoon peaks are observed for Participants 4, 8, 11 and 18, indicating daily routines centered around these times, while Participants 11 and 8 stand out with evening peaks, likely due to evening household activities. Participant 11 shows the most intense consumption, suggesting a higher energy-dependent lifestyle, whereas Participant 12 displays isolated usage, indicating infrequent high-energy activities. These patterns highlight a spectrum of energy behaviors, from stable low-consumption users to those with distinct peak demands. Interestingly, the energy consumption pattern of participant 13 aligns with sunrise and sunset times, as this participant has installed photovoltaic (PV). Other participants such as as 4, 10, 14 but their usage is not as consistent as participant 13 except may participant 4.

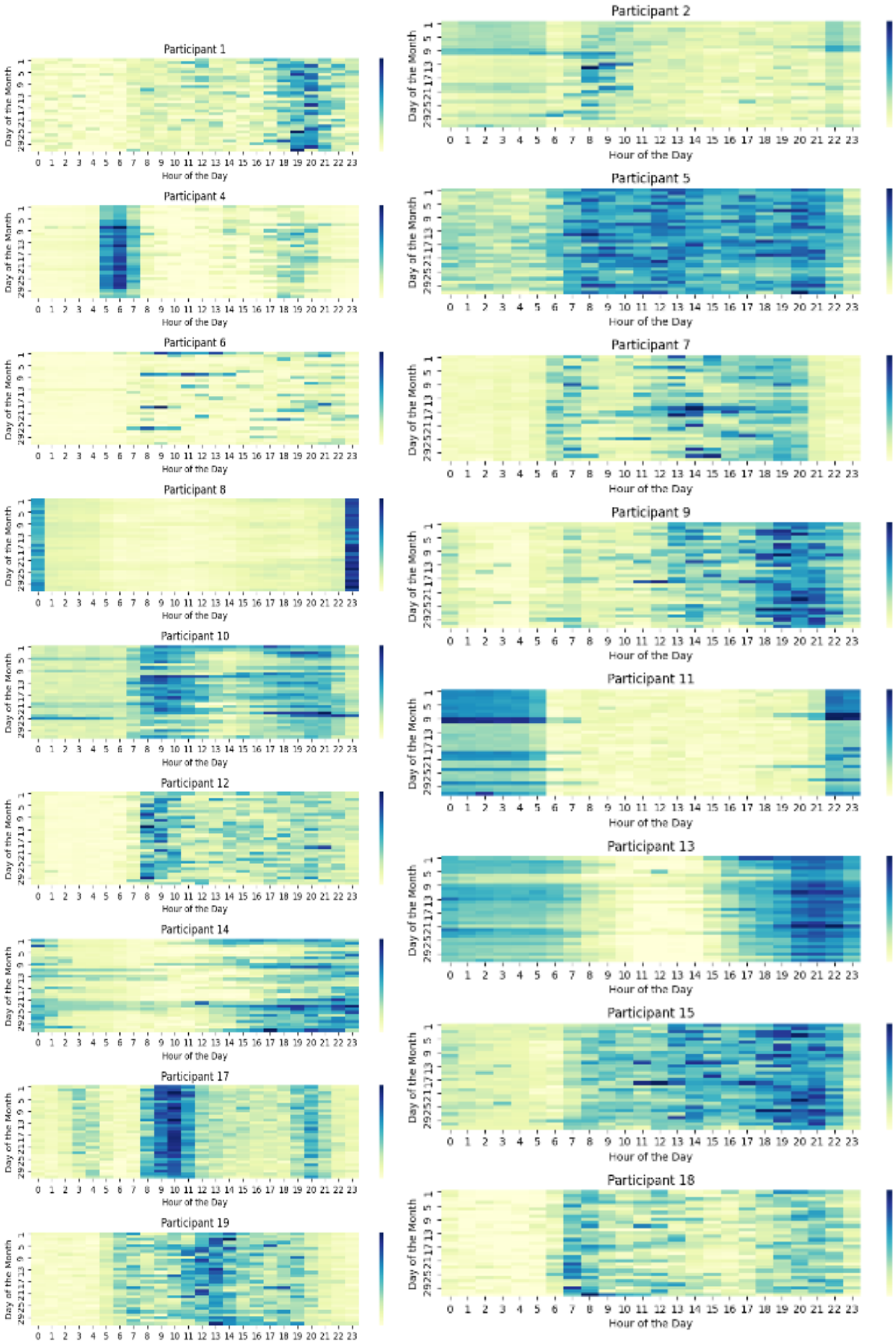

Figure 30: Participants Electricity Consumption Profile

4.2.3 Baseline Modeling - Electricity Demand Prediction

A key objective of this project is to predict energy demand—in this case, electricity consumption—for the targeted energy community. For this purpose, we utilized data from the Maritrost energy community, which includes electricity consumption records of individual participants. Given the data's structure, we approached this task with a multi-model learning strategy, creating separate models for each participant to capture the unique consumption patterns of each. The rationale behind this approach is that each participant exhibits distinct characteristics in their consumption profile, independent of others within the community. Since we only have individual consumption data and lack other contextual information (like demographic or appliance data), a personalized model for each participant can be trained using only the time-series data. This approach can avoid the need for cross-participant feature engineering, simplifying the modeling process. Additionally, each participant's model can be tuned specifically to their behavior, leading to more precise predictions. This learning framework helps with optimizing load predictions and energy management strategies for each user, which is valuable for both participants and the community as a whole.
We implemented a simple baseline data-driven (machine learning) model for predicting energy demand. Baseline models are typically designed with a minimal parameter space required for fine-tuning. For each participant, the model follows a one-step forecasting approach, predicting consumption at time *t+1* based on the current consumption at time *t*. In other words, it involves developing a function *f(x)* that maps the input variable *x* (energy consumption at time *t*) to minimize the average prediction error over training instances for forecasting consumption at *t+1*. Once trained for each participant, the model is applied iteratively on the test set, meaning each prediction is fed back as input to forecast subsequent steps.

In the initial experiments we trained *n* models (estimators) where *n=number of participants*. The underlying model is classical machine learning method called linear regression. The data is resampled to hourly sum of energy consumption for each participant, hence the prediction is done for hourly energy demands. The learning function for linear regression is as follows:

$$y_{t+1} = \beta_0 + \beta_1 x_t + \epsilon$$

The model learns the parameters β0 and β1 by minimizing the mean squared error (MSE) between the predicted values yt+1 and the actual values at t+1 over the training data:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_{t+1}^{(i)} - (\beta_0 + \beta_1 x_t^{(i)})\right)^2$$

This error minimization allows the model to find the best fit line for predicting t+1 values based on t-step values in the training set. Once trained, this function can be used to predict future values iteratively.

The evaluation of the baseline model is performed through 10-fold cross validation for each participant. Figure 21 illustrates the Mean Squared Error (MSE) distribution for training and testing across different participants in an electricity demand prediction model.

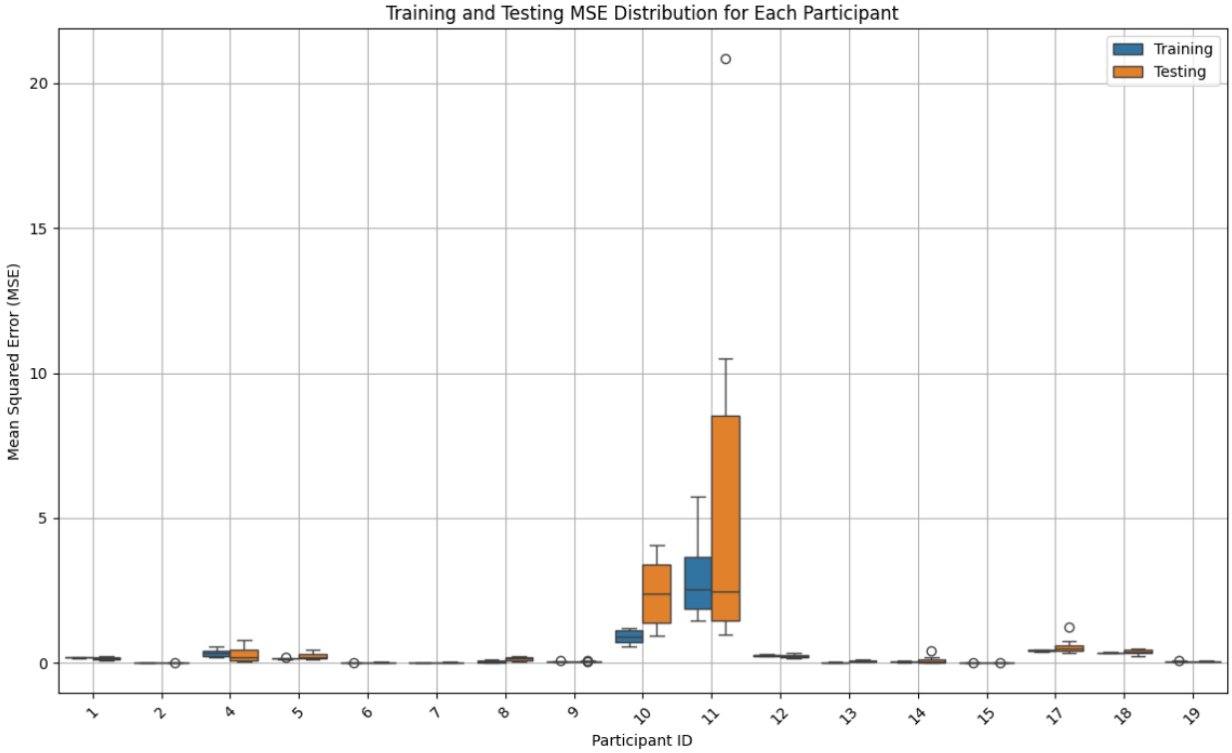

Figure 31: Average MSE for Baseline Model for Hourly Energy Demand Prediction

The majority of participants show low and consistent MSE values for both training and testing, indicating that the model generally performs well in capturing demand patterns. However, participants with IDs 10 and 11 exhibit significantly higher MSE, particularly in the testing phase, with a noticeable spread and outliers, suggesting that the model struggles to generalize effectively for these individuals. This discrepancy aligns with the findings and analysis presented in Figure 20 that indicate unique or irregular demand patterns for these participants, which the model fails to predict accurately. To address this issue, additional information will be integrated into the learning feature space to enhance the model's effectiveness. One potential approach involves incorporating non-operational variables, such as weather data, obtained from open-source APIs. This data will be collected for the periods corresponding to the Energy Community's measurement timeline and made available for analysis. Nonetheless, initial findings suggest that electricity consumption alone serves as an effective indicator, and the model demonstrates sufficient capacity for predicting energy demand in the baseline setup. Additionally, other baseline models, including support vector machines, XGBoost, random forests, and advanced deep learning methods, will be adapted and trained. Following this, data-driven learning methods and strategies will be developed to assess and compare performance improvements over the baseline model.

4.3 Austria – Campus Inffeldgasse load prediction for mixed use districts (DiLT)

Integrating increasingly distributed energy resources alongside a diversifying range of loads introduces significant volatility to existing energy systems. Accurate load prediction emerges as a foundational element for balancing supply and demand. We investigate data-driven methods for short- and long-term energy load prediction on a district scale for mixed-use districts that include high-energy-consuming facilities, using the example of Campus Inffeldgasse at TU Graz.

The university campus comprises different types of buildings and usages, ranging from offices, study halls, classrooms, gastronomy, and labs. The operator defined the requirements for the prediction model, which includes predictions ranging from one hour to several weeks to test various services on the campus. A particular challenge is the

unpredictable energy consumption patterns from high energy demand labs, where single experiments can produce loads. While these loads appear to be purely stochastic from a historical data analysis point of view, they are planned and scheduled events in the real world. This scheduling information is usually available before the actual experiments are undertaken, as the electricity network operators have to be informed with due notice of the MW scale load surges.

The campus consists of 30 buildings totalling 124.000 square meters of net floor area (Figure 32). These buildings and floor spaces are used for offices, lecture halls, gastronomy, and labs. The annual energy consumption of the campus is currently 18 GWh of electricity, 11 GWh of heat, and 3 GWh of cooling. Additionally, 966 kWp of photovoltaic systems are installed on the campus.

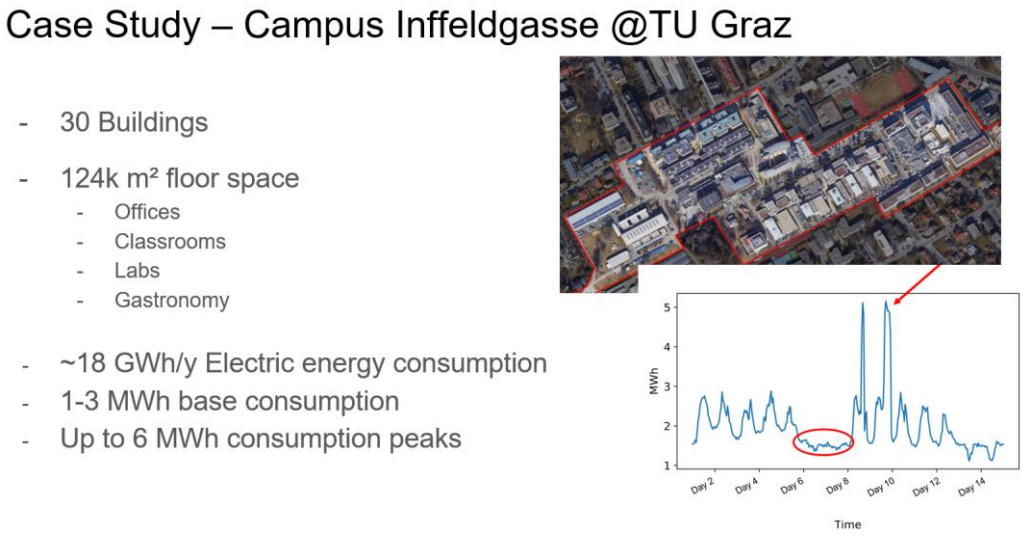

Figure 32: Key facts about the Case Study

Figure 33 shows two weeks of the total electrical energy consumption of the Inffeldgasse campus. We can observe day-based and week-based periodicity, with the lowest power consumption on the weekends. Normal load/operation can be observed in the first week at around 3 MW peak and 2 MW during nights. Special loads - in our case big labs, e.g., a substantial-size turbine test rig - can be observed in the second week on Monday and Tuesday, leading to a steep increase in power consumption peaking above 5 MW. These special loads mostly occur during business hours and are active only for a few hours. From a pure data perspective, these special loads behave highly stochastic and are therefore impossible to predict with classical methods lacking further information. However, the actual experiments in these labs and test rigs have to be planned and scheduled beforehand - going as far as the energy provider for the campus has to be notified with due diligence about such load surges. This work investigates if this scheduling information can improve energy consumption prediction and its impact on short- and long-term prediction.

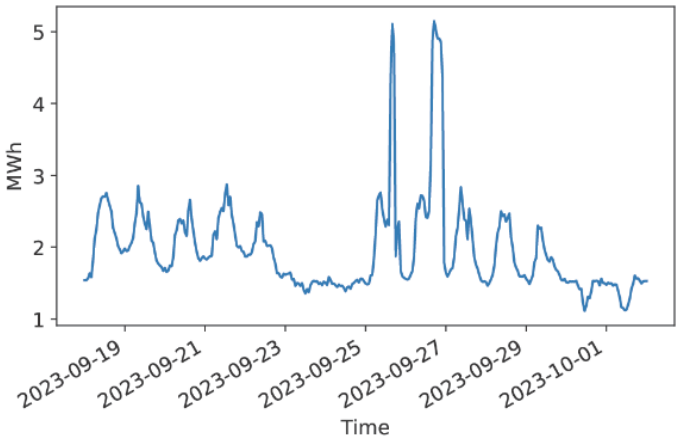

Figure 33: The total energy consumption of the Inffeld campus over two weeks.

4.3.1 Dataset

The dataset used in this experiment consists of energy consumption data from the Inffeldgasse campus from May 2022 until January 2024, as well as weather forecast data for Graz, Austria for the same period. The data is presented in hourly intervals, resulting in 11732 data points. Initial analysis of the data shows that common load profiles for the campus are: Between 1 and 3 MW for regular building operation, with the lowest loads around Christmas (between 1 and 2 MW), and peaks beyond 6 MW due to special loads.

While scheduling information for heavy-load laboratories does exist, it is distributed over multiple management tools and platforms. As this work only investigates the feasibility of using this information for prediction, connecting these data sources and automating the collection of this data is out of scope. We therefore created the scheduling information from the nine biggest consumers from their standalone power consumption available in the dataset. Table 1 shows a short description of these nine installations and their maximum loads that occur within our dataset.

Table 1: Description of laboratories and other heavy loads for which scheduling information is utilized.

Lab	Peak (kWh)	Connected devices
1/9	372	Server room; Heat pumps (cooling + heating)
2/9	390	Large engine center (test hall); Hydrogen research center
3/9	430	Server room; Heat pumps (experiment); Climate chamber
4/9	2426	Thermal turbo-machinery (big compressors)
5/9	2256	Server room; Heat pumps (cooling)
6/9	2003	Thermal turbo-machinery (high pressure compressors)
7/9	2003	Heat pumps (cooling + experiment)
8/9	2365	Heat pump (cooling + experiment); Roller test bench; Engine test bench bench
9/9	365	Heat pumps (cooling + heating); Clean room

Whenever the power consumption of a laboratory/special load exceeds its mean by more than 0.15 times its standard deviation, we assume the special load to be active. The resulting binary vectors form our representation of laboratory/special loads scheduling information.

The data collected within this study and all sourcecode of our models and evaluations are made publicly available on Zenodo [1].

4.3.2 Models

The models used can be split into two categories, based on the approach to the problem, utilized feature sets, and temporal scope. The first category is formed by short-term prediction models that predict the next time step's energy consumption in a recurrent manner based on the previous values. The second consists of long-term prediction models that predict the consumption for any given time in the future, based only on general information.

The base features used in both model categories are:

- Workday - Binary value being 1 for Monday until Friday, and 0 for Saturday and Sunday.
- Class day - Binary value being 1 for workdays during the semester, and 0 for weekends, holidays, and lecture-free periods.
- Day of Week - Cyclic encoding of the day of week (0-6).
- Month - Cyclic encoding of the month (0-11).
- Time of day - Cyclic encoding of the time in hours (0-23).
- Temperature - Outside temperature in degrees Celsius.
- Lab Schedule - Binary value for each of the nine laboratories being 1 if the special load is in use and 0 otherwise.

Short-Term Prediction Models
The short-term models have the last N hours of energy consumption as additional features and predict the consumption of the next hour.
Based on our learnings from preliminary experiments, we set $N=2$ as a look-back window size.
To perform predictions over longer periods, the models are applied recurrently, with the prediction for time T being fed back into the model as a feature to predict the next timestamp $T+1$.
The weather information (temperature) used in this class of models is based on weather forecasts obtained at $T=0$.

Long-Term Prediction Models
The long-term models use only the base features described above to predict the energy consumption for an arbitrary point in the future.
The weather information (temperature) used in this class of models is based on climate data collected from past years and represented as monthly means.

Ensemble Model
We select the best-performing model from each category, short- and long-term, and use their outputs as features for a custom Voting Regressor model producing a final prediction.
The input of this Voting Regressor consists of the predictions of the short-term model for a period of $N=150$ hours starting at time T, and the predictions of the long-term model for the same period predicted from start time T.

ML Methods
We investigated the application of XGBoost (default parameters), Linear Regression (default parameters), Elastic Net (alpha=0.8), Decision Tree (max_depth=10), Random Forest

(max_depth=10, n_estimators=100), Bayesian Ridge (default parameters), Artificial Neural Network (ANN) with a single hidden layer of width 16, and Deep Neural Network (DNN) with three hidden layers each 16 neurons wide for this prediction task. NN-based models are trained for 50 epochs. We evaluated the performance of each of these methods in both, the short-, and long-term approaches. Min-Max scaling is applied to all input features.

Furthermore, we perform SHAP analysis to investigate the impact and importance of all our features. We implemented our experiments in Python 3.9 and did not perform hyperparameter tuning, relying on the default parameters provided within the scikit-learn (v1.4.0) and XGBoost (v2.0.3) libraries. The neural network models are implemented using Keras (v2.10.0) and TensorFlow (v2.10.0), and the applied model architecture and size are based on existing literature and established rules of thumb [2].

4.3.3 Evaluation

We performed time-series cross-validation [3, 4] with five folds to evaluate model performance on our prediction task. The available data is ordered by time and then split into folds, where training is performed on folds *F1..Fn* and the performance of the model is then measured on fold *Fn+1* to avoid information leakage. The performance scores reported in the results section are means of the resulting *n-1* folds evaluation scores. We employ R2 [5], Mean Average Percentage Error (MAPE) [4], Mean Squared Error (MSE) [6] and Mean Average Error (MAE) [4] as performance metrics. Two trivial baselines are introduced to compare the achieved performance of our models: The repeat window baseline (i) uses a repeated pattern of energy consumption from the past 24 hours as the prediction; The last value baseline (ii) takes the last known energy consumption value for the whole prediction horizon.

4.3.4 Results and discussion

Short-Term, Recurrent Models Performance:
Table 2 shows the R2 scores of cross-validation for a prediction horizon of up to five hours. The linear regression and random forest model perform best for predicting the energy consumption of the upcoming hour. As to be expected, model performance decreases towards longer prediction horizons. This is due to the prediction error accumulating as the output of the model for one prediction step is fed back into the model as a feature for predicting the next step. Beyond a prediction horizon of five hours, linear regression is outperformed by the DNN. The random forest performs best overall and is chosen for further experiments.

Table 2: Table 2. Mean R2 scores from time-series cross-validation of short-term prediction using recurrent model inference.

Model	**Prediction horizon in hours**				
	1	2	3	4	5
XGBoost	0.90	0.85	0.81	0.78	0.76
Linear Regression	0.91	0.85	0.81	0.79	**0.79**
Elastic Net	0.46	0.43	0.41	0.41	0.41
Decision Tree	0.84	0.79	0.75	0.72	0.69
Random Forest	**0.92**	**0.88**	**0.85**	**0.82**	**0.79**
Bayesian Ridge	0.91	0.85	0.81	0.79	**0.79**
ANN	0.88	0.83	0.80	0.79	0.78
DNN	0.91	0.85	0.82	0.80	**0.79**

Short-Term, Recurrent Models, With and Without Special Load Inclusion:
Figure 34 shows the energy consumption prediction when including or excluding lab schedule information from the feature set. We can observe that the models integrating heavy load scheduling information strictly outperform the models without. Furthermore, the prediction performance of models without scheduling information very quickly deteriorates to impractical levels, dropping below $R2 \leq 0.5$ within 4 hours, while the models incorporating this information continue performing at or above $R2 \geq 0.7$. We conclude that models substantially benefit from the scheduling information for special loads.

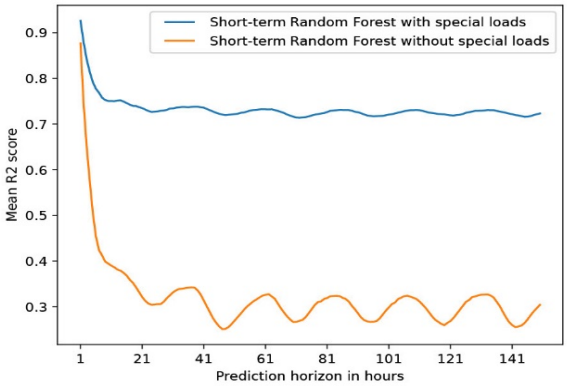

Figure 34: Short-term models based on Random Forest with and without special loads as features over a prediction horizon of 150 hours.

The recurrently operating short-term models perform well for short prediction horizons, but towards growing prediction horizons they are outperformed by the long-term models. Figure 35 shows the R2 scores of the best performing short- and long-term models (both based on Random Forest). We can observe that beyond a prediction horizon of four hours, the R2 score of the short-term, recurrent, model falls below the long-term model and continues falling.

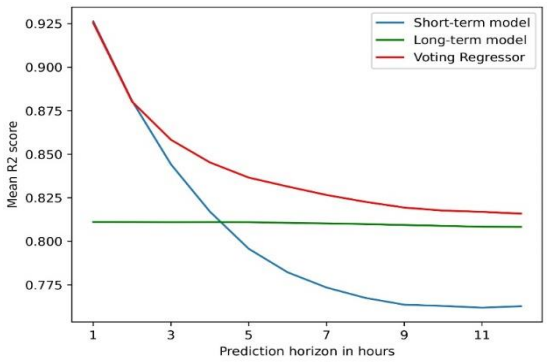

Figure 35: Comparison of the best performing short- and long-term models (both Random

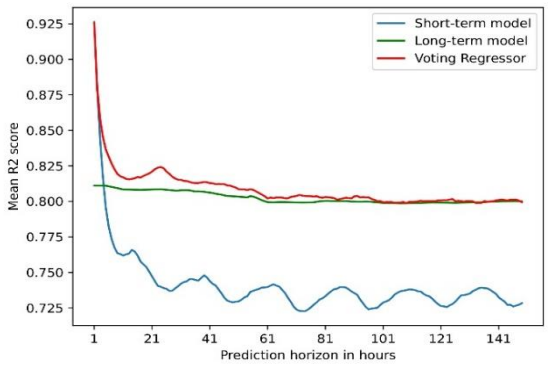

Figure 36: R2 scores from cross-validation of the Voting Regressor, short- and long-term models over a prediction horizon of 150 hours.

Short- and Long-Term, Ensemble Model Performance:
A Logistic Regression model is used as a Voting Regressor to combine the benefits of the short- and long-term models. Figure 35 and 36 show the R2 scores of the Voting Regressor together with the underlying short- and long-term models performance. The performance of the Ensemble across the transition point, where the long-term model is starting to outperform the recurrent model, is substantially better than what can be achieved by switching short-term and long-term models depending on the current prediction horizon ($R2=0.85$ vs. $R2=0.82$ for a 4-hour prediction). Also, for mid- and long-term predictions

after the transition point, the Voting Regressor performs better than the long-term model alone (*R2=0.81* vs. *R2=0.80* for a 72-hour prediction).

Figure 37 shows the cross-validation MAPE scores for these models together with the trivial last value baseline and repeat window baseline for a prediction horizon of 96 hours. MAPE score is chosen because the R2 scores of the trivial baselines are too bad for a reasonable comparison plot. We can observe that the trivial baselines perform consistently and substantially worse than all our models.

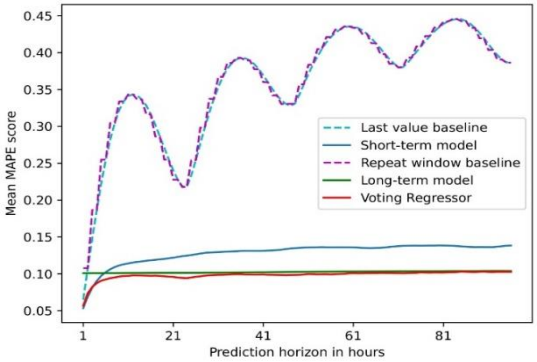

Figure 37: MAPE scores from cross-validation of the Voting Regressor, short- and long-term models and the trivial baselines over a prediction horizon of 96 hours.

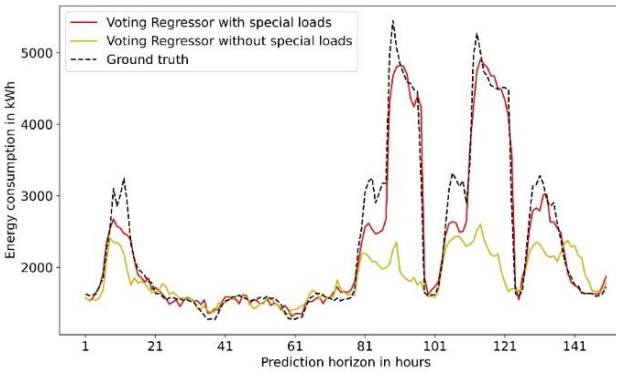

Figure 38: Energy prediction using the Voting Regressor with and without lab schedule information for the next 150 hours.

Figure 38 shows the prediction result of our Ensemble for a 150-hour prediction horizon with, and without lab schedule information. Again, we can observe that load prediction without scheduling information is infeasible due to the stochastic nature of these heavy loads. Peaks are even predicted for points in time where no peak exists.

Feature Importance Analysis

We performed SHAP feature importance analysis on the Random Forest based long-term prediction models.

To do so we split the whole dataset into train and analysis set in an 80/20 ratio and trained our models on the former split, followed by applying SHAP on the latter split. Figure 39 shows the mean SHAP values of the top ten features from the feature set used in our long-term models. The importance of scheduling information is obvious from this analysis, with these features for the biggest energy consumers (Lab 2/9 and Lab 4/9) being more impactful than any other feature. Time of day and information if the day in question is a working day, a weekend, or a holiday, are more important than the actual day of the week for which load prediction is performed. The small impact of outside temperature as a feature can be explained by heating systems on the campus being mostly fed from other sources than electrical energy. Features outside the top ten are listed in descending order of their resulting SHAP values: Hour (sin), lab 5/9, day of week (sin), lab 9/9, lab 3/9, class day, month (cos), and day of week (cos).

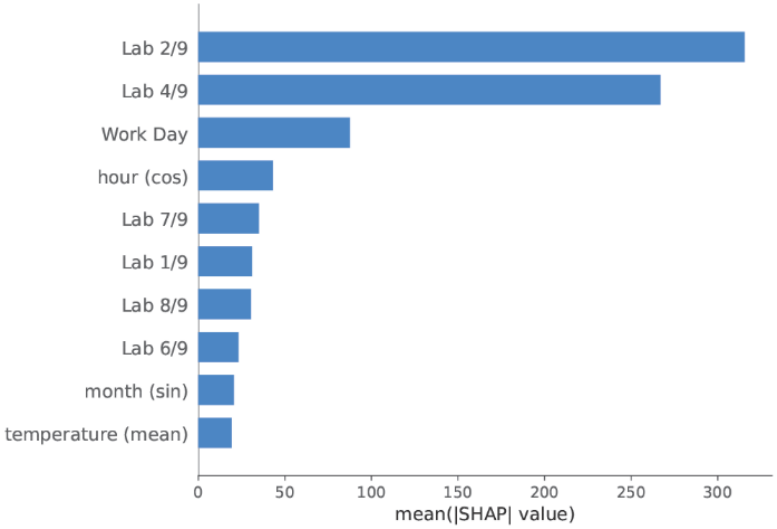

Figure 39: Top ten features of SHAP analysis of Random Forest performing long-term prediction.

4.3.5 Conclusion

In this paper, we investigated the incorporation of heavy load lab scheduling information into ML-based short- and long-term load prediction models for mixed-use districts. We show that load scheduling information can significantly increase load prediction performance from *R2=0.37* to *R2=0.75* for a twelve-hour prediction horizon. As expected, short-term prediction methods relying on past time steps load information perform very well for prediction horizons of a few hours (*R2=0.92* for one hour, and *R2=0.82* for four hours). However, performance then quickly declines below the performance scores of the long-term models (*R2=0.80* for arbitrary prediction horizons) as errors accumulate in the recurrent invocation of the short-term models. We therefore implemented an ensemble approach to combine the benefits of both short- and long-term prediction models while mitigating their limitations. The applied ensemble approach can provide a single point of contact and outperforms both standalone models' prediction performance by achieving a score of *R2=0.85* and *R2=0.81* for a 4-hour and 72-hour prediction respectively.

5 Fault detection and diagnostics (TU Wien)

5.1 One-class Classification Cluster ENsembles (OCCEN) for Fault Detection and Diagnosis (TU Wien)

This study introduces a new framework for fault detection and diagnosis in energy-efficient buildings, focusing specifically on air handling units, and tests it using synthetic data. The framework is structured to identify faults and provide relevant diagnoses without requiring prior knowledge of specific faults during the modeling process, employing an unsupervised learning approach. This approach intersects with physics-informed AI, particularly in non-linear dynamic systems and industrial contexts. Physics-informed AI guides model learning by enforcing the physical system's defined boundaries and flags data points as anomalies when they exceed these constraints without explicit annotations of anomalies during training. This research is published at an international conference on Principles of Diagnosis and Resilient Systems (DX24). A brief overview of the work is provided below.

5.1.1 Motivation and Introduction

Real-world automated systems such as building automation, power plants, and more have benefited from data-driven learning methodologies for anomaly detection and diagnosis. Typically, these methodologies heavily rely on prior knowledge related to abnormal operations, i.e., data points labeled as anomalies. However, in practice, such labelled data points are often unavailable which poses challenges in effective anomaly detection, particularly in diagnosis. In this paper, we propose One-class Classification Cluster ENsembles (OCCEN) anomaly detection and diagnosis approach for multivariate time series data. OCCEN utilizes one-class classification learning methods for anomaly detection followed by the decomposition of anomalies into multiple clusters. Then each cluster is treated as a binary classification problem and classifiers are trained to learn cluster representations. These trained models in combination with explainable AI models are used to generate a ranked list of diagnoses, i.e., features. Finally, we re-rank those features to account for temporal dependencies through the dynamic time-warping technique. The practical evaluation of OCCEN for air handling units (AHU) demonstrates its effectiveness in identifying faults. The framework consistently outperforms the baseline in fault diagnosis, as higher scores are observed for detection and diagnostic evaluation metrics, including F1 score, intersection over union, HitRate@k, and RootCause@k.

5.1.2 Framework and Methodology

The overall framework of OCCEN is presented in Figure 40. For the task of anomaly detection, the objective is to learn a function f : T -> {0, 1} where f(t_i) = 0 represents a non-anomalous instance and f(t_i) = 1 indicates anomalous data point. Furthermore, it is assumed that characteristics of T represent normal operations, and deviations from normal patterns are indicative of faults.

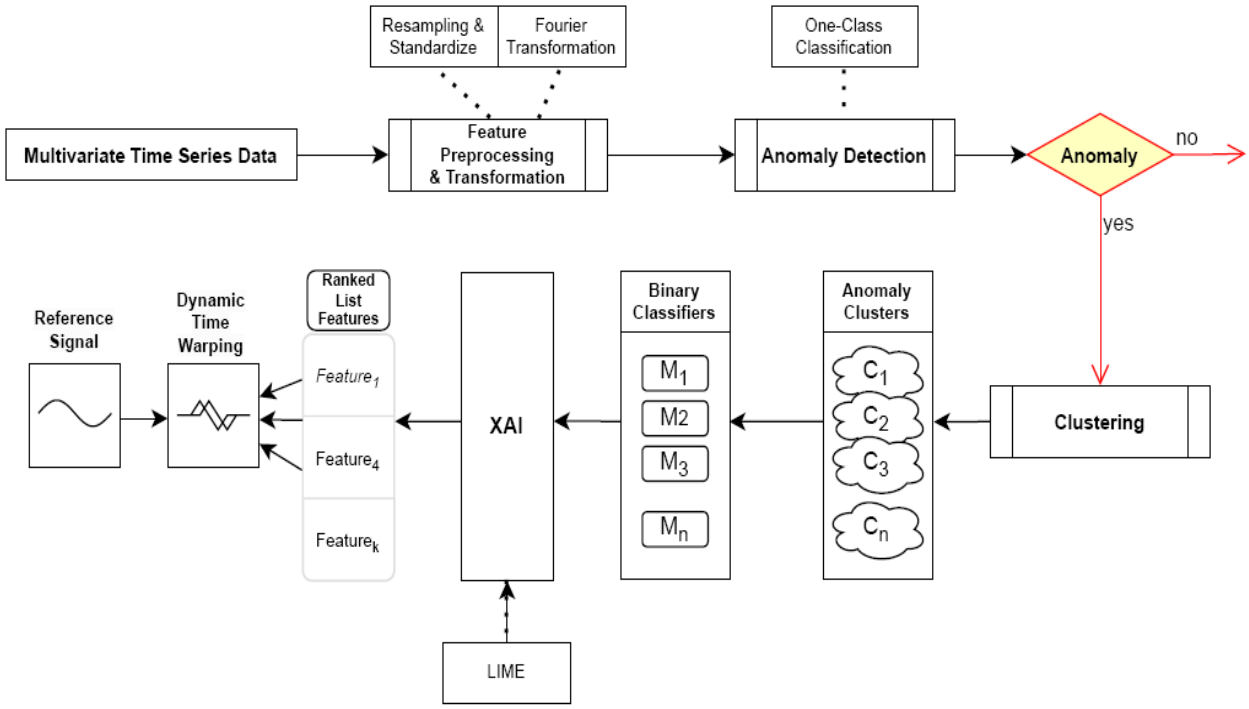

Figure 40: Fault Detection and Diagnosis Framework

Labeling large datasets, particularly high-resolution time series data, is very challenging in real-world situations because it requires a lot of time and effort. The sparsity of class labels (minority class), e.g., anomalies in our case, in the datasets negatively affects the learning

capability and performance of the model due to the learning bias towards the dominating class labels (majority class). To address this, one-class classification (OCC) methods are often used. One-class classification methods are a special case of binary- or multi-classification learning methods. These methods are extensively used for detecting anomalies and novelties in time series data. The main concept involves learning the distribution of a single class, typically normal observations, and establishing decision boundaries that differentiate between inliers and outliers. For the anomaly detection task, various One-Class Classification models are evaluated and the best one is considered to generate a ranked list of diagnosis through XAI and Dynamic time warping technique. The algorithm code for the diagnosis part is presented in Figure 41.

Explainable AI methods such as LIME and SHAP are often considered useful to explain the decision made by the classifier. This is accomplished by generating a ranked list of features that are indicative of the decision. These features are considered the most influential in terms of decision-making of the underlying base model. In this work, we use explainable methods as the first step toward the task of feature ranking for the anomalies. We identify the most significant features for each anomalous object in the time series. This is done by assigning the anomalous object to a particular cluster and then using the relevant binary classifier associated with that cluster, as the base model for the explainable AI method. So far, feature generation occurs at a specific time step t, and time series data often includes temporal dependencies that can impact the effectiveness of fault diagnosis. To address this, we take into account a set of features generated by explainable methods over a time sequence spanning a length of m. However, a longer time sequence will result in a large number of diagnoses, i.e., features, and it may affect the fault diagnosis performance. Therefore, to further pinpoint the diagnoses, we rely on the FastDTW to rank the features based on point-wise distance.

5.1.3 Dataset

In this study, we employ a synthetic dataset of a single-duct air handling unit (AHU), published by Granderson et al. The dataset comprises one year of operational data, incorporating both faulty and nominal system behaviors, with a total of 525,600 time samples representing each operational condition. Faults are imposed on sensors and control sequences at various biases (see Figure 41), but our diagnostic approach identifies fault types without focusing on bias levels. We train OCC models for anomaly detection using only normal operations data. Pre-processing includes resampling to hourly averages, applying Fast Fourier Transformation (FFT), and feature scaling with the min-max method. These steps, along with trained models, are applied to preprocess test data, ensuring no data leakage.

Fault Type	Fault Annotation	Method of Fault Imposition
Supply air temperature sensor bias	Sa_temp	Add a bias value to the sensor output
Outdoor air temperature sensor bias	*Oa_temp*	Add a bias value to the sensor output
OA damper stuck	*Dmpr_stk*	Automated override of outdoor air damper position to indicate it is stuck
Cooling coil valve leaking	*Cvlv_lkg*	Adjusted the coil valve position value when the control signal is zero
Cooling coil valve stuck	*Cvlv_stk*	Override of the coil valve position to indicate that the valve is stuck

Figure 41: Summary of Fault Imposition Methods for AHU

Algorithm 1 Diagnosis through Anomaly Detection, Clustering, and Feature Ranking

Require: T: Set of all instances (both normal and anomalous)
Require: $f(t_i)$: Anomaly detection function (1 if anomalous, 0 if normal)
Require: E: Explainable model (e.g., LIME)
Require: m: Window length for feature retention
Require: T_{normal}: Set of fault-free (normal) observations
Ensure: Ranked list of critical features for root cause diagnosis

Step 1: Anomaly Detection
$F \leftarrow \{\}$ ▷ Set of all detected anomalies
for each $t_i \in T$ **do**
  if $f(t_i) = 1$ **then**
    $F \leftarrow F \cup \{t_i\}$
  end if
end for

Step 2: Clustering
Apply clustering C on F: $C(F) = \{c_1, c_2, \ldots, c_n\}$

Step 3: Train Binary Classifiers for Each Cluster
for each cluster $c_j \in C(F)$ **do**
  $D_{c_j} \leftarrow$ instances in c_j
  $D_{\text{other}} \leftarrow$ randomly select $|D_{c_j}|$ instances from all other clusters $i \neq j$
  $D_{\text{train}} \leftarrow D_{c_j} \cup D_{\text{other}}$
  Train binary classifier B_j on D_{train}
end for

Step 4: Feature Explanation and Ranking
for each cluster $c_j \in C(F)$ **do**
  for each instance $t_i \subset D_{c_j}$ **do**
    $F_{\text{top}}(t_i) \leftarrow$ top-ranked features from $E(t_i)$
    Retain $F_{\text{top}}(t_i)$ over a window of length m
  end for
end for

Step 5: Calculate FastDTW Distance for Top Features
for each feature $f_r \in F_{\text{top}}(t_i)$ **do**
  $DTW_{\text{distance}}(f_r, T_{\text{normal}}) \leftarrow \text{FastDTW}(f_r \text{ in faulty sequence}, f_r \text{ in } T_{\text{normal}})$
end for

Step 6: Rank Features by FastDTW Distance
Rank features in $F_{\text{top}}(t_i)$ based on $FastDTW_{\text{distance}}(f_r, T_{\text{normal}})$ in descending order

Step 7: Apply Truncation
Truncate the ranked feature list to retain only the most critical features with the highest FastDTW distances
return Truncated list of critical features for root cause diagnosis

Figure 42: Diagnosis through Anomaly Detection, Clustering, and Feature Ranking

5.1.4 Baseline Method & Evaluation Metrics

We evaluated the efficacy of different OCC methods in detecting anomalies and considered the best-performing method for the fault diagnosis analysis. To assess fault diagnosis performance, we established a simple baseline by employing explainable AI models immediately following the OCC methods, unlike OCCEN. The explainable model generates a list of ranked features, i.e., diagnoses, for a given time step. The top features ranked by the XAI model are then retained for over m time steps and further re-ranked based on distance measures calculated by FastDTW.

The design of the baseline approach is shown in Figure 43. It is based on the premise that explainable AI models are generally good at identifying the key features that affect the

decisions of the base model and FastDTW technique will further improve the ranking. The objective is now to assess how well this baseline performs in feature ranking compared to OCCEN.

In our study, each fault type has a single root cause, but in non-linear dynamic systems, one failure can impact multiple components, making it crucial to examine these effects to identify the root cause. We consulted building automation experts to review the dataset and list possible diagnoses for each fault type. To evaluate fault diagnosis performance, we adapted two metrics: ***Overlap@P*** and ***HitRate@k***. ***Overlap@P*** measures the overlap between true diagnoses and the top *P x* |*GT*| ranked diagnoses, with *P* as 150% or 200% and |*GT*| as the number of true diagnoses. Most fault types have 5 true diagnoses, except Sa_temp with 2, capping ranked diagnoses at 10 when *P* is 200%. We also used ***HitRate@k***, which checks if at least one true diagnosis is within the *top k* ranked diagnoses. Our dataset includes five fault variants: Sa_temp, Oa_temp, Dmpr_stk, Cvlv_lkg, and Cvlv_stk. To measure the accuracy of identifying root causes, we introduced ***RootCause@k***, assessing the proportion of instances where the true cause is ranked among the top k diagnoses. We compared the effectiveness of the baseline and OCCEN using these metrics across sliding window sizes from 3 to 24 for diagnosis performance.

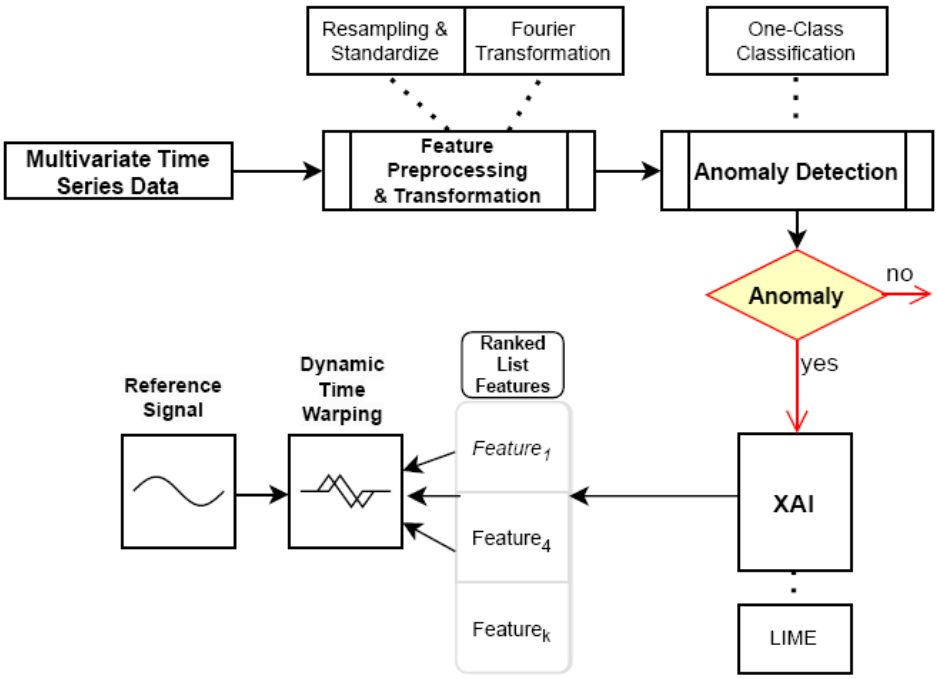

Figure 43: Baseline Approach Design

5.1.5 Results

Fault Detection: The average results of the anomaly detection task are presented in Figure 44. The Elliptic Envelope model consistently shows high precision and recall across all fault types, resulting in strong F1 scores, particularly in detecting Cooling Coil Valve Stuck (Cvlv_stk) anomalies. Its precision above 93% and recall above 90% in most cases reflect its robustness and reliability in anomaly detection.

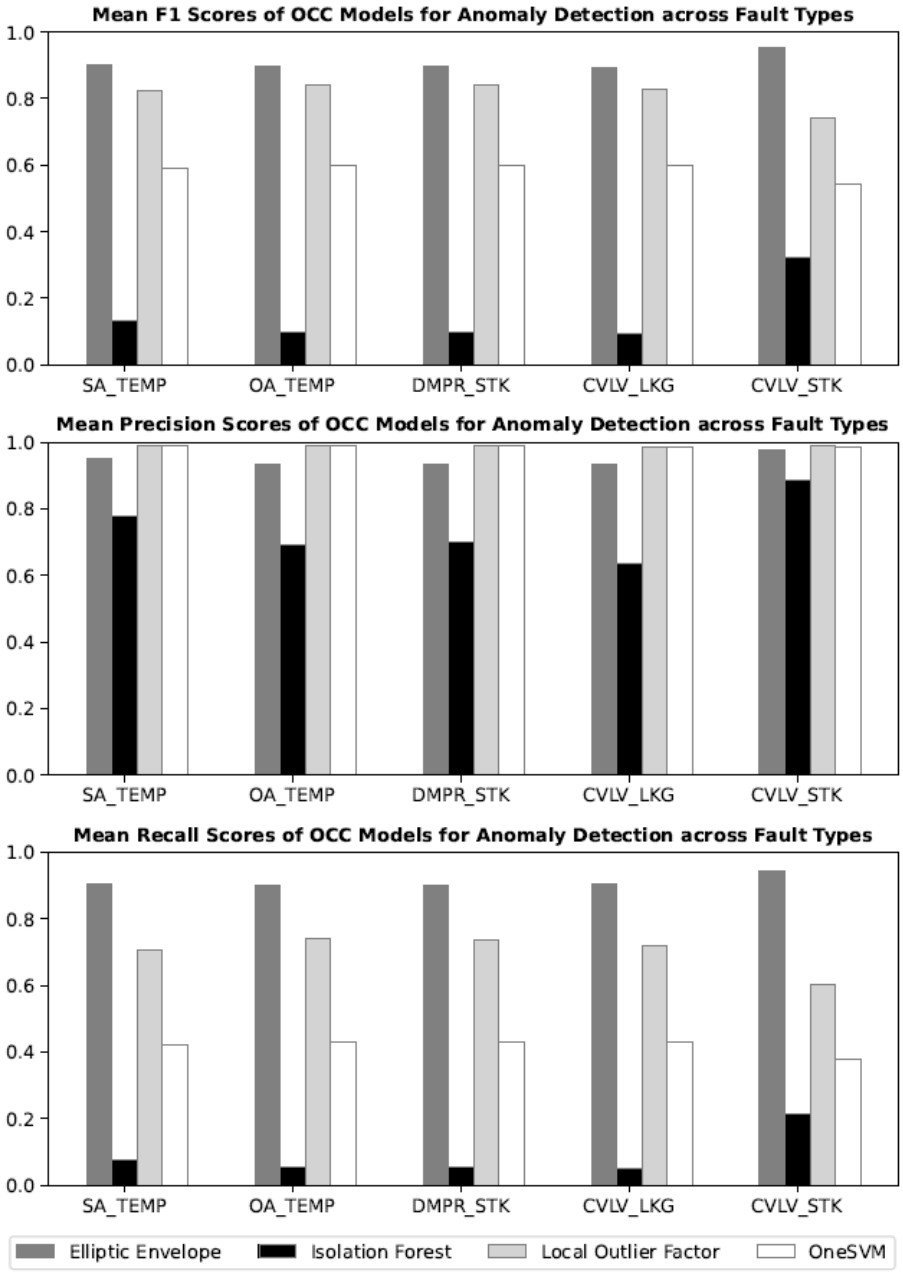

Figure 44: 10-Fold Cross Validation Evaluation

Fault Diagnosis:

***Overlap@P*:** The baseline and OCCEN comparative analysis is presented in Figure 45. Overall, OCCEN consistently performs better than the baseline at ranking relevant diagnoses for all fault types and achieves higher overlap percentages. This becomes more pronounced for larger window sizes.

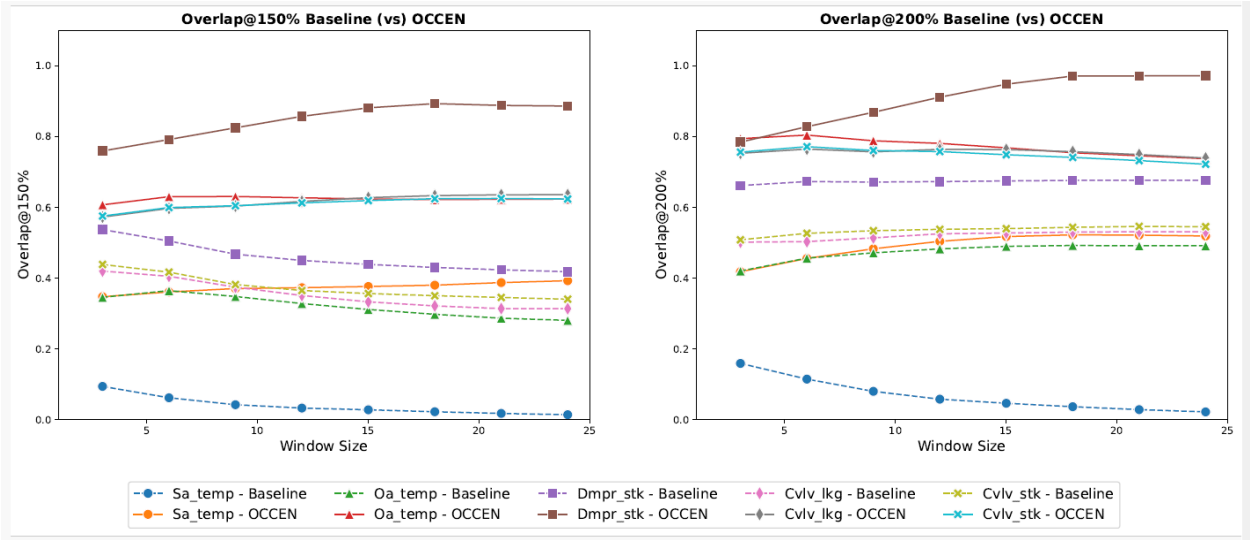

Figure 45: Mean score comparison of Overlap@P metric for baseline and OCCEN

HitRate@K: The overall evaluation of HitRate@k metric is presented in Figure 46. Overall, the results demonstrate clear performance improvements for OCCEN in identifying at least one true diagnoses in top k ranked diagnoses and it improves as window size increases. Conversely, the baseline shows, in general, lower hit rate scores but performs relatively better for large window size.

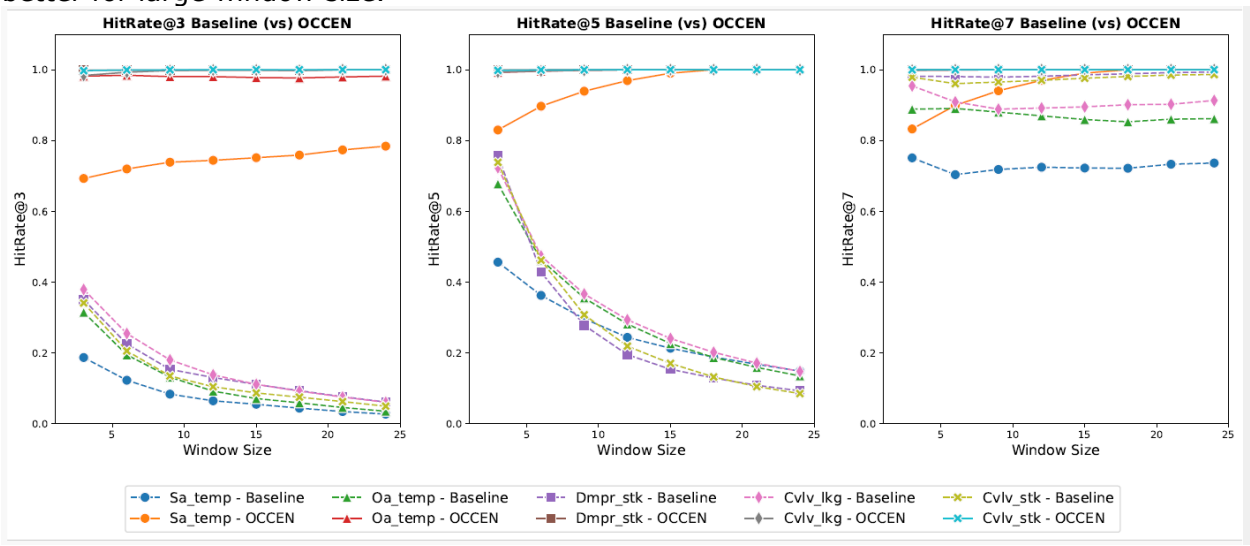

Figure 46: Mean score comparison of HitRate@k metric for baseline and OCCEN

RootCause@k: Remember, in this study, each fault type has only one true cause. Therefore, in this evaluation procedure we are particularly interested in measuring how accurately the true fault cause is identified within the top-k ranked diagnoses. Figure 47 provides an overview of the outcomes and at first glance it can be observed that OCCEN performs better at ranking true diagnoses than the baseline.

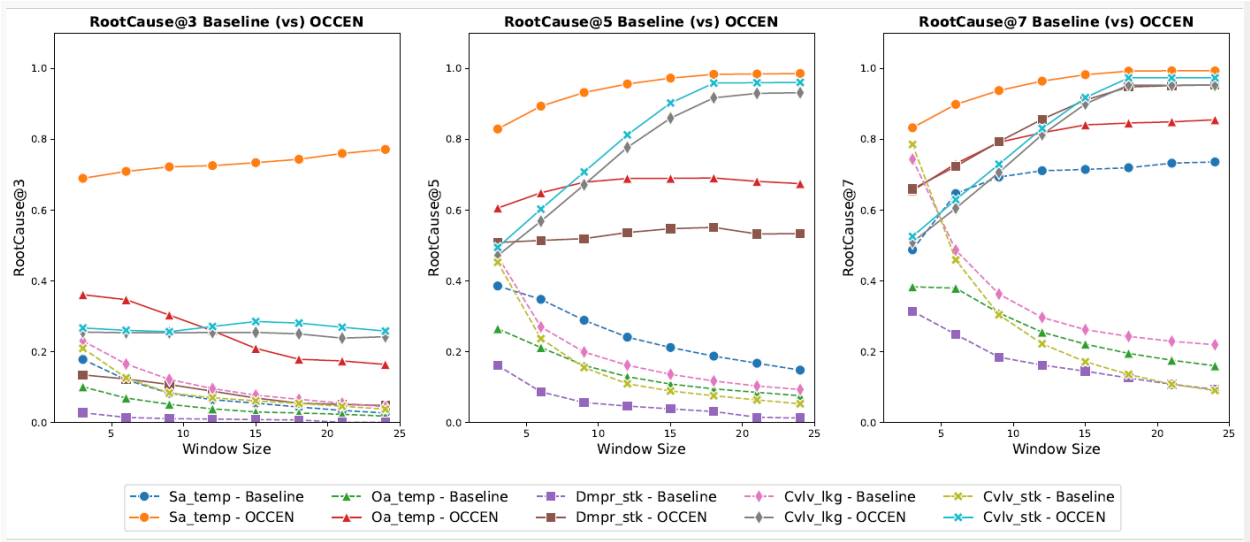

Figure 47: Mean score comparison of RootCause@k metric for baseline and OCCEN

5.1.6 Conclusion & General Remarks

In this study, we developed and proposed an anomaly detection and diagnosis framework called OCCEN. OCCEN relies solely on non-anomalous instances of multivariate time series. Learning cluster representation of anomalies, where each cluster can represent a fault type, the framework enhances feature significance and associations within each cluster. A ranked list of diagnoses is produced for each cluster assignment (i.e., anomaly) by employing the XAI method (LIME) and the sequence matching technique (FastDTW) to consider temporal dependencies. The extensive evaluation of OCCEN on a synthetic dataset of the real-world use case of an air handling unit demonstrated that OCCEN outperforms the classical baseline method.

The work and findings presented in this report are briefly describe for the brevity of understanding and to limit the number of pages. The work is published in DX24 conference.

5.2 Fault Detection and Diagnosis for Air Handling Units (RWTH)

According to the World Energy Outlook [7] report from 2025, the building sector accounted for 28% of global total final energy consumption. Enhancing operational efficiency within the built environment is a cornerstone of modern energy transition strategies. This is particularly relevant for energy communities, where collective energy management and grid stability depend on the predictable and efficient performance of participating assets. The RWTH trial site addresses this by focusing on fault detection and diagnosis for the central air handling unit (AHU) of a test hall. Research indicates that approximately 40% of AHUs suffer from active faults at any time [8]. These malfunctions significantly degrade both energy efficiency and occupant thermal comfort [9]. The integration of FDD mechanisms and subsequent maintenance protocols is critical to achieve energy savings. Moreover, FDD transforms building assets into reliable, efficient nodes within an energy community, thereby supporting broader decarbonization goals and enhancing the flexibility of the local energy network

5.2.1 Use Case

The test hall (Figure 48) was commissioned by the Institute for Energy Efficient Buildings and Indoor Climate of the E.ON Energy Research Center (E.ON ERC). The building is partitioned into two functional areas: one dedicated to research and testing of energy system components, and the other serving as an office workspace. The central AHU (Figure 39) provides conditioned air to these areas, utilizing jet nozzles for the test hall and active chilled beams for the office. Although the test hall is also equipped with concrete core activation, the FDD focuses exclusively on the central AHU, given the AHU's more direct role in maintaining occupant comfort. The central AHU can provide heating, cooling, as well as humidification and dehumidification to both zones.

Figure 48: Test Hall building for the RWTH use case

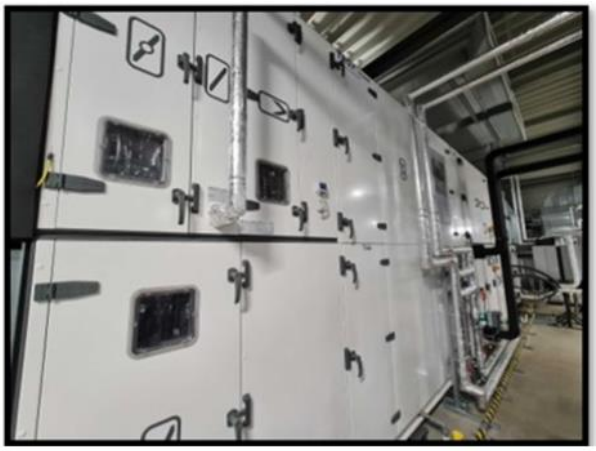

Figure 39: Air handling unit in the test hall

The schematic layout of the AHU is illustrated in Figure 49, detailing a configuration that includes four dampers, two fans, a heat recovery system, a steam humidifier, and a set of heat exchangers comprising two heating coils and one cooling coil. The developed FDD focuses on the heating and cooling coils, which are integrated via identical hydronic mixing circuit, as detailed in the schematic shown in Figure 50. The district heating/cooling station supplies a heating/cooling distributor. The temperature coming from the respective distributor is the primary water-side supply temperature ($T_{W,\mathrm{Sup},Prim,C}$). A mixing valve then uses this primary water-side supply and the water-side return temperature ($T_{W,Ret,C}$) to modulate the required water-side supply temperature ($T_{W,\mathrm{Sup},C}$).

To standardize the nomenclature, variables are defined using T (temperature), $\dot{V}$ (volume flow), n (rotational speed), M(operating mode), and u (valve position). These are indexed by medium Air (Air) or Water (W), spatial location (Supply: Sup, Return: Ret, Primary: Prim), and setpoints (Set). The specific component C is denoted as follows: preheater (PH), cooler

(CO), or reheater (RH). An exception to this convention is the pump setpoint ($P_{Set,C}$), which varies based on the operational mode.

The preheater functions as a frost-protection safety unit, activating whenever ambient temperatures drop below 8°C. Its operation is defined by a constant-speed pump, with the water supply temperature ($T_{W,Sup,PH}$) managed via a PID-controlled mixing valve (u_{PH}). In contrast, the reheater (RH) and cooler (CO) employ a specialized control logic: the valve position governs the water-side supply temperature ($T_{W,Sup,C}$), while the pump speed is utilized to regulate the air-side return temperature ($T_{Air,Ret,C}$). This dual-control approach aims to optimize pump efficiency, a strategy further elaborated by Teichmann et al. [10].

Control logic is executed on a programmable logic controller (PLC). Communication occurs via a hybrid pathway: some signals are sent directly to components using the BACnet protocol, while others pass through a manufacturer-specific PLC. Due to the proprietary nature of this manufacturer's logic, certain underlying control processes remain opaque and cannot be independently validated. All operational data, encompassing sensor readings, actuator setpoints, and system modes, is monitored and stored in a database and retrieved in 5-minute intervals via an API for analysis.

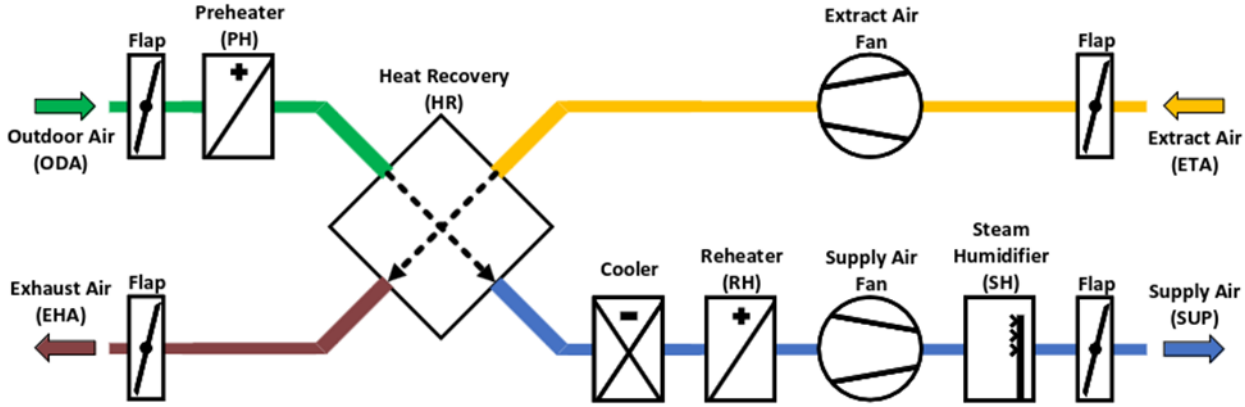

Figure 49: Schematic of the air handling unit and its components [11]

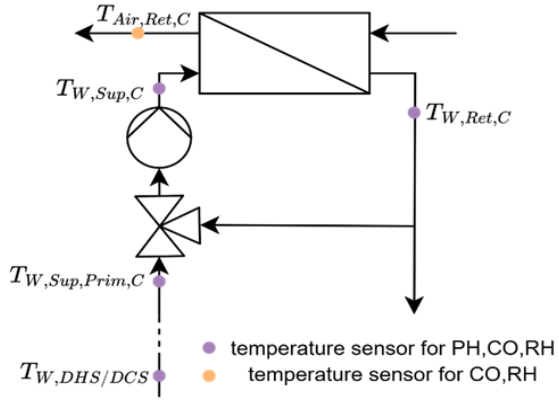

Figure 50: Schematic representation of the investigated heating and cooling coils and their hydraulic mixing circuit.

5.2.2 Rule-based Hierarchical Fault Detection and Diagnosis Methodology

This section details the proposed FDD framework, which utilizes an automated loop designed to identify and isolate root-cause faults at each discrete time step. The process commences by retrieving relevant time-series data from the monitoring system. For every heat exchanger identified within the system topology, a hierarchical, rule-based diagnostic routine is executed (see Figure 51). This logic employs a cascading structure: each block

represents a distinct fault category, and subsequent checks are conditional upon the detection of a preceding fault. Diagnostics are systematically sequenced from symptomatic indicators, such as deviations in supply air temperature, toward the underlying root causes, with specific criteria outlined in Figure 52. These rules are aligned with the specific requirements of the current system configuration. Beyond the current implementation, the modular design of this framework supports future further automation. The use of standardized middleware, such as FIWARE, could standardize data exchange and simplify integration into wider energy community networks. The execution of the loop concludes with a log documenting the identified fault types for each time step.

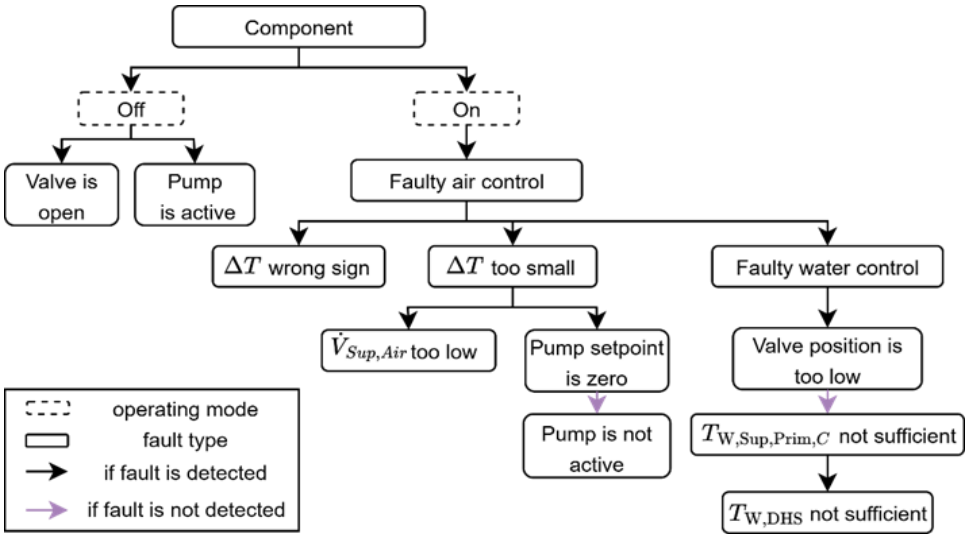

Figure 51: Hierarchical order of the fault types for the preheater, cooler, and reheater. The rules are derived from the control logic and the hydraulic circuit.

Fault type	Rules
Faulty air control	$T_{\mathrm{Air,Ret,Set},C} - 1\ \mathrm{K} < T_{\mathrm{Air,Ret},C} < T_{\mathrm{Air,Ret,Set},C} + 1\ \mathrm{K}$
ΔT too small	$\mid T_{\mathrm{W,Sup},C} - T_{\mathrm{W,Ret},C} \mid < 0.5\ \mathrm{K}$
ΔT wrong sign	$T_{\mathrm{W,Ret},CO} - T_{\mathrm{W,Sup},CO} < 0, T_{\mathrm{W,Sup},C} - T_{\mathrm{W,Ret},C} < 0\ \forall C \in [\mathrm{PH}, \mathrm{RH}]$
Faulty water control	$T_{\mathrm{W,Sup,Set},C} - 1\ \mathrm{K} < T_{\mathrm{W,Sup},C} < T_{\mathrm{W,Sup,Set},C} + 1\ \mathrm{K}$
$\dot{V}_{Sup,Air}$ too low	$\dot{V}_{Sup,Air} < 0.1$
Pump setpoint is zero	$P_{\mathrm{Set},C} = 0\ \%$
Valve position is too low	$u_{\mathrm{Set},C} < 0.1\ \% \vee\ u_C < 0.1\ \%$
Pump is not active	$n_{\mathrm{Pump},C} < 0.1 \wedge \dot{V}_{\mathrm{Pump},C} < 0.1$
$T_{\mathrm{W,Sup,Prim},C}$ not sufficient	$T_{\mathrm{W,Sup,Set,CO}} > T_{\mathrm{W,Sup,Prim,CO}}, T_{\mathrm{W,Sup,Set},C} < T_{\mathrm{W,Sup,Prim},C}\ \forall C \in [\mathrm{PH}, \mathrm{RH}]$
$T_{\mathrm{W,DHS}}$ not sufficient	$T_{\mathrm{W,Sup,Set,CO}} > T_{\mathrm{W,DCS}}, T_{\mathrm{W,Sup,Set},C} < T_{\mathrm{W,DHS}}\ \forall C \in [\mathrm{PH}, \mathrm{RH}]$
Valve is open	$u_{\mathrm{Set},C} > 0.1\ \% \vee\ u_C > 0.1\ \%$
Pump is active	$P_{\mathrm{Set},C} > 0.1\% \vee\ n_{\mathrm{Pump},C} > 0.1 \vee \dot{V}_{\mathrm{Pump},C} > 0.1 \vee M_{\mathrm{Pump},C} \neq 0$

Figure 52: Detection rules for fault types

5.2.3 Use case rule-based FDD results for the water-air heat exchangers

In the following subsections, the results for the different water-to-air heat exchangers are shown. The results for preheater and reheater are presented from the 01.10.2025 to the 31.12.2025 and the results for the cooler are shown from the 01.07.2025 to the 30.09.2025.

5.2.3.1 Results for the Preheater

The results of the fault detection and diagnosis for the preheater alongside its operating modes are shown in a heatmap in Figure 53. During this period, several recurring fault patterns were identified. For instance, the fault *Valve is open* is frequently detected during the preheater's off-mode periods. Furthermore, a cluster of three simultaneous faults occurred toward the end of October 2025. Two of these, *Faulty water control* and *Valve*

position is too low, are linked hierarchically, as defined in Error: Reference source not found, and are therefore analyzed in conjunction. Similarly, the fault types *ΔT too small* and *ΔT wrong sign* are treated as concurrent events. It is also observed that *Faulty water control* is triggered almost continuously while the preheater is in heating mode.

Although the heatmap provides significant insights into the system's performance, a detailed evaluation of the preheater results is omitted here, as these findings are analyzed extensively in a separate conference paper "Knowledge graph-enabled rule-based fault detection and diagnosis for air handling units" at ECOS 2026. By accurately detecting operational inefficiencies, sensor errors, and system failures, the proposed FDD methodology proves to be effective for the preheater. The subsequent analysis focuses on the remaining system components and the verification of the FDD methodology using selected time intervals.

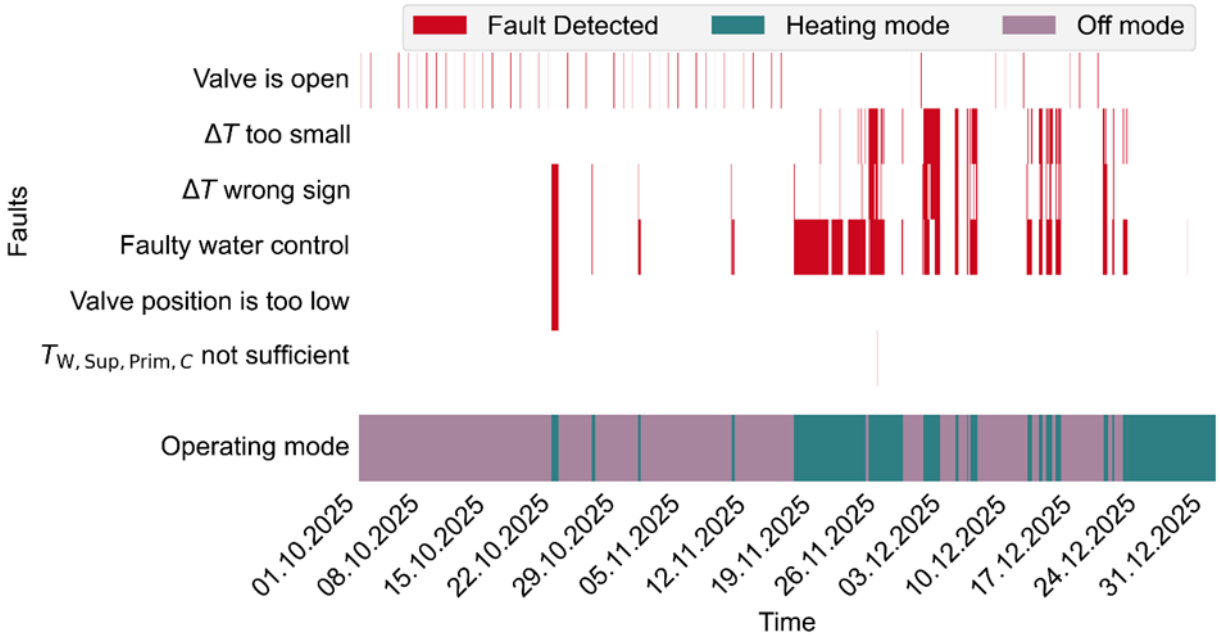

Figure 53: Heatmap of all fault types of the preheater occurred from the 01.10.2025 to the 31.12.2025 and operating mode of preheater. To facilitate visualization, data is aggregated into 1-

5.2.3.2 Results for the Cooler

Figure 54 displays a heatmap of the cooler's fault detection results and operating modes. In the heatmap four different fault combinations are unconcealed. First, *valve is open* fault which occurs mostly at the end of *faulty air control* faults but once occurs standalone during a period when the cooler is off. This period is analyzed first. Moreover, at the beginning of July, one fault combination appears, including nearly all shown fault types, and is thus analyzed together. The fault $T_{W,Prim,C}$ *not sufficient* occurs always during dehumidification mode. This fault type is hierarchically connected to the *faulty water control* and *faulty air control* fault.

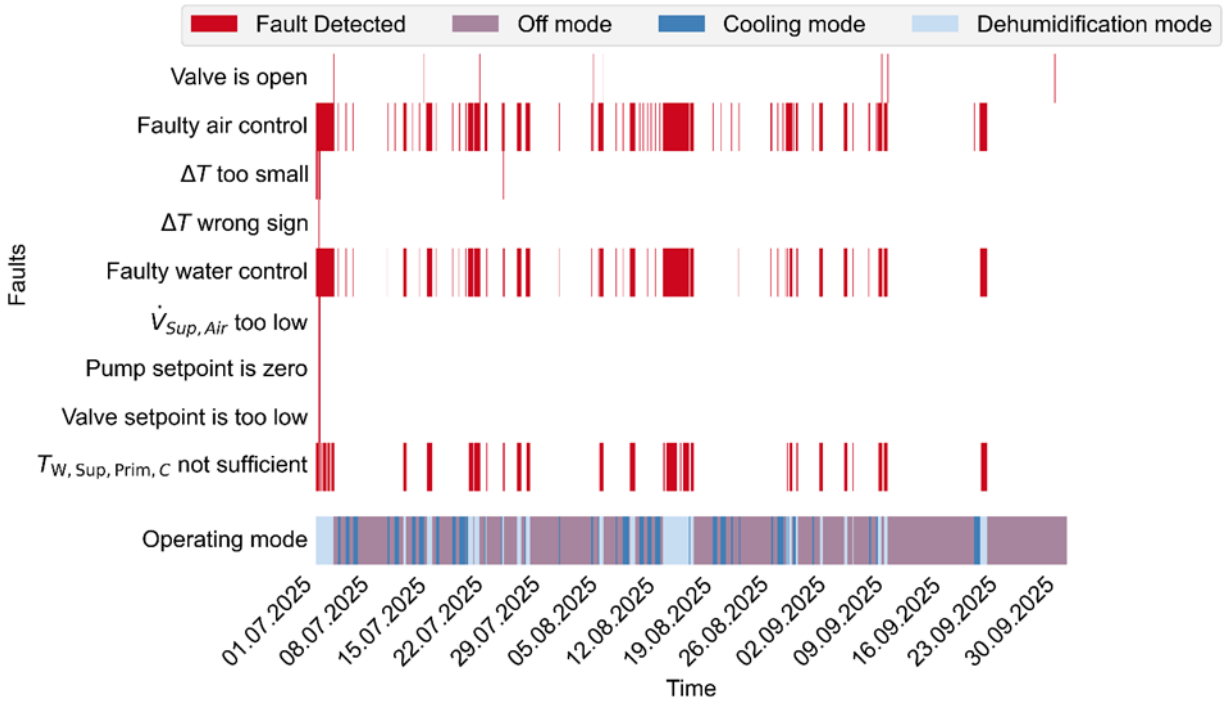

Figure 54: Heatmap of all fault types of the cooler occurred from the 01.07.2025 to the 30.09.2025 and operating mode of cooler. To facilitate visualization, data is aggregated into 1-hour bins.

For the detected fault *Valve is open* during an off-mode period, the valve position and the operating mode are shown in Figure 55. Generally, the valve setpoint is given between 0-100%. Unlike the pump setpoint, the actuation signal of the valve is not sent directly from the supervisory control to the valve. The actuation signal is first sent to the manufacturer's PLC. This PLC remodulates the setpoint between ~30-73%. Around 2pm, the valve opens ~1% more than usual for around 5 min, which should not happen while the preheater is off and the setpoint is zero. Unfortunately, due to communication issues the pump datapoints are not available during this period. But the setpoint of the pump is zero and it should be off during off mode of the cooler. Therefore, the valve's opening despite a zero setpoint indicates internal leakage, likely caused by pressure difference in the supply and return flow of the cooling distributor.
In Figure 56 the fault detection of the fault *faulty air control* is shown for a period where only this fault is detected, together with relevant measurements and the operating mode of the cooler. The air-side temperature first drops too low at the beginning of each cooling period. Unfortunately, due to communication issues, the pump measurements are not available during the whole plotted period, but the setpoint (percentage of the speed) of the cooler pump is available. This setpoint reaches its PLC limited maximum of 80% during the beginning of the fault, which causes the high temperature drop. Due to a slow reaction of the PID controller, the air-side temperature only reaches its setpoint after 0.5-1 hours. This issue could be mitigated by tuning the PID parameters, which would allow the system to reach the air-side setpoint faster and subsequently reduce the pump's setpoint faster. After these 0.5-1 hours, the pump operates at the minimum percentage setpoint (0%), which equals a speed of 400 rpm. This pump speed is sufficient to reach the air-side temperature setpoint. The fault *faulty air control* is correctly detected as the initial fault, as the tuning of the PID could solve the fault.

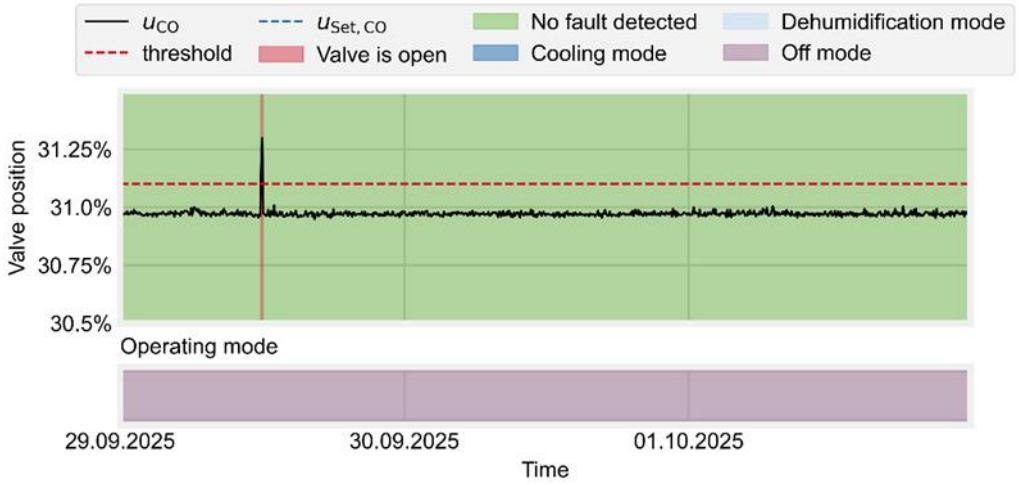

Figure 55: Fault detection and diagnosis results and operation mode for the fault type "Valve open" for the time period from 29.09.2025 to 01.10.2025.

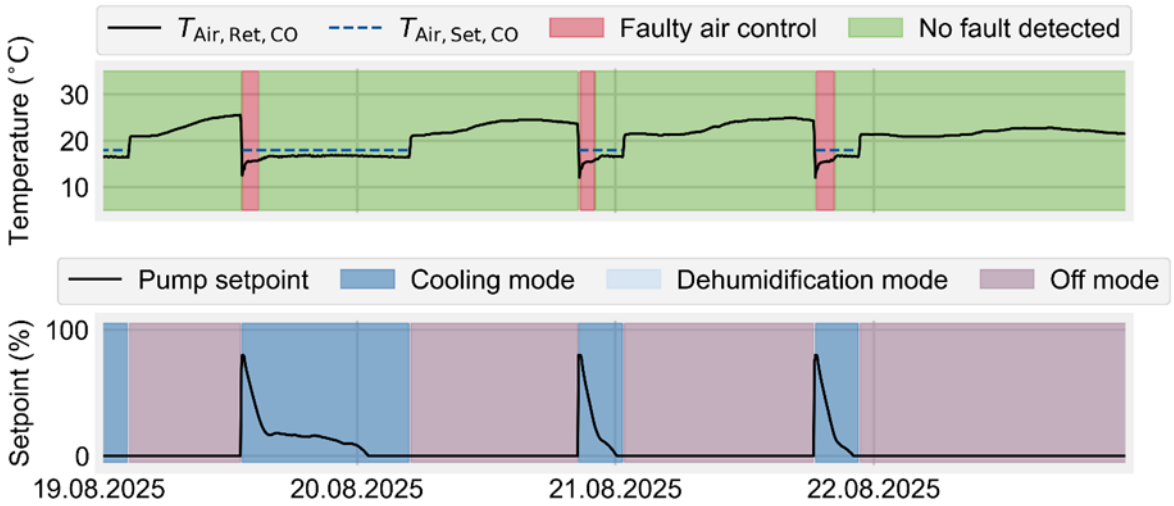

Figure 56: Fault detection and diagnosis results and operation mode for the fault type "Faulty air control" for the time period from 19.08.2025 to 22.08.2025.

In this paragraph the hierarchical combined faults, *faulty air control, faulty water control* and $T_{W,Prim,C}$ *not sufficient* are analyzed. In Figure 57 the fault detection results, relevant measurements, and the operating mode are shown for the period 19.09.2025 – 21.09.2025. At the beginning of the faulty period, the fault *faulty water control* is shortly detected as the water-side supply temperature needs some time to reach the setpoint. Right afterwards, the fault *faulty air control* is detected during a short period at the beginning of the cooling mode. The reason for this is described in the paragraph above. It is noteworthy to mention that in this period, it takes ~2.5 hours to reach the air-side temperature setpoint. During the dehumidification mode, the fault *faulty air control* is detected. The fault is always detected if the cooler is in dehumidification mode, together with the hierarchical connected faults *faulty water control* and $T_{W,Prim,C}$ *not sufficient.* The pump is operating at its maximum speed during this period. It cannot be set to a speed higher than 80%, as noises that indicate severe abrasion occur with speeds above this threshold. Moreover, the valve setpoint is at its maximum of 100%, which translates to a real valve opening of 73%. The cause of the fault is the insufficient primary supply temperature, which is higher than the setpoint. Even if the mixing valve is fully opened to the primary supply to eliminate recirculation of the return, the resulting supply temperature ($T_{W,Sup,CO}$) still fails to reach the required setpoint. But a higher valve opening is not possible due to the remodulation in the

manufacturer's PLC. The initial detectable fault is therefore $T_{W,Prim,C}$ *not sufficient.* The fault $T_{W,DCS}$ *not sufficient* is not triggered and thus the temperature provided by the district cooling network is sufficient to provide a cooler primary water-side supply temperature. It is not clear why $T_{W,Prim,CO}$ is not low enough.

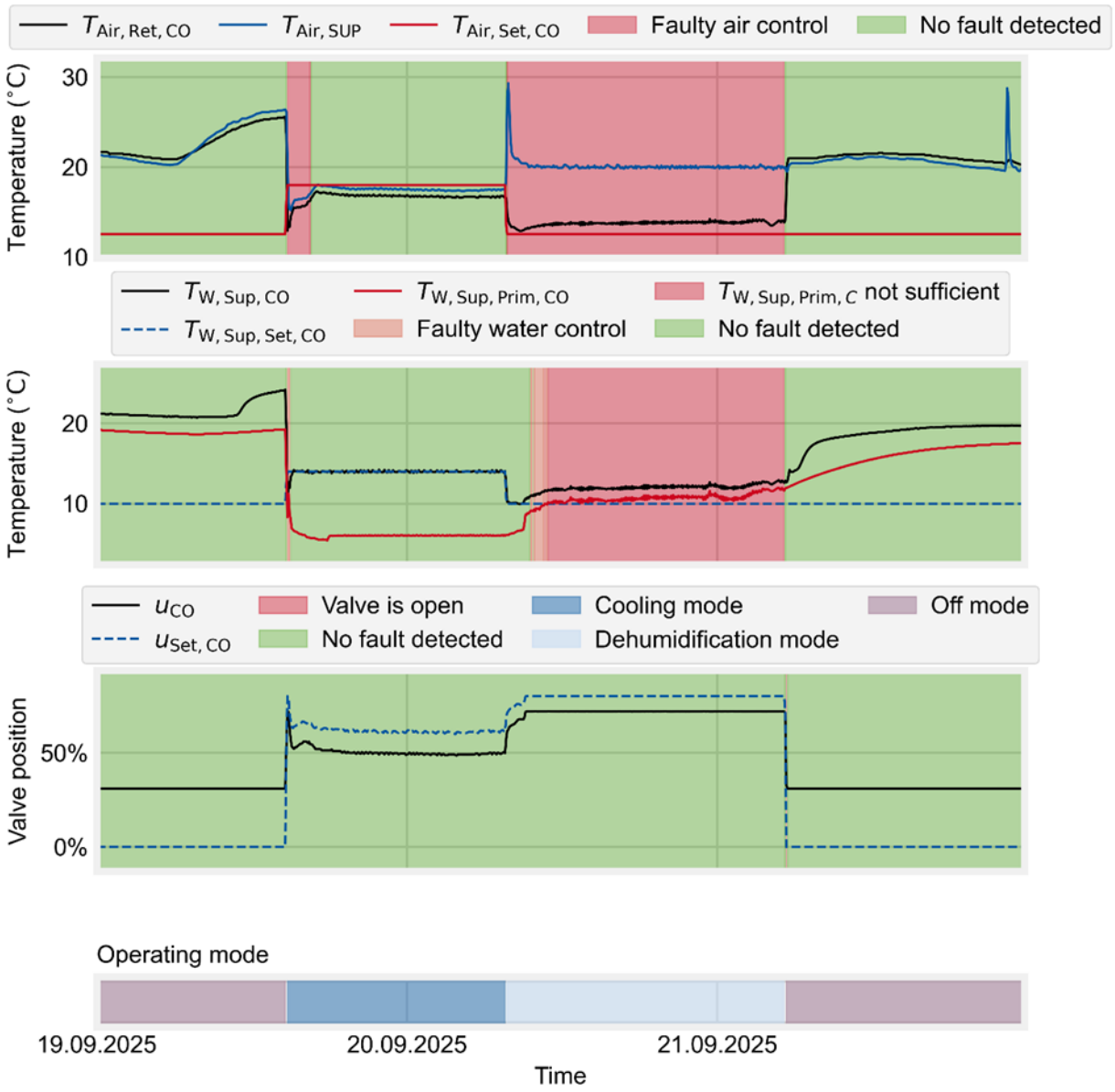

Figure 57: Fault detection and diagnosis results and operation mode for the connected fault types "Faulty air control", "Faulty water control" and "T_W,Sup,Prim,C not sufficient" for the time period from 19.09.2025 to 21.09.2025.

In Figure 58 and Figure 59 the fault detection results for the combined faults, which occur at the beginning of the investigated three summer months, relevant measurements and operating modes for this period are displayed. During the beginning of the displayed period the following faults occur simultaneously: faulty air control, faulty water control, $T_{W,Prim,C}$ not sufficient and Delta T too small. The fault $T_{W,Prim,C}$ not sufficient is usually the initial fault for both control faults during dehumidification mode, as described above. The fault Delta T too small here also could be a root cause for the faulty air control, as with a too small temperature difference not enough heat can be transferred to the air-side and the air temperature setpoint cannot be reached. Nevertheless, all sensors were calibrated in the winter 2025/26 and offsets were found for the water-side temperatures. Therefore, the fault

most likely is not a real fault but rather indicates wrong calibrated sensors. This indication will be checked in the cooling period during spring/summer 2026. After this first fault combination, the faults *valve setpoint is too low,* $\dot{V}_{Sup,Air}$*too low* and *Pump setpoint is zero* are the initial occurring faults. The AHU experienced a scheduled multi-hour shutdown of the fan, pump, and heat exchanger valve to allow for preheater maintenance. Due to the absence of upstream filtration, the preheater is susceptible to progressive clogging, requiring routine cleaning. Consequently, the fault detection system flags this downtime, as the normal operational sequences are interrupted and a deviation from the expected setpoints is present. Following these two new fault patterns, the typical not sufficient during dehumidification mode and the faulty air control in the beginning of a cooling period are occurring. These were already described in detail.

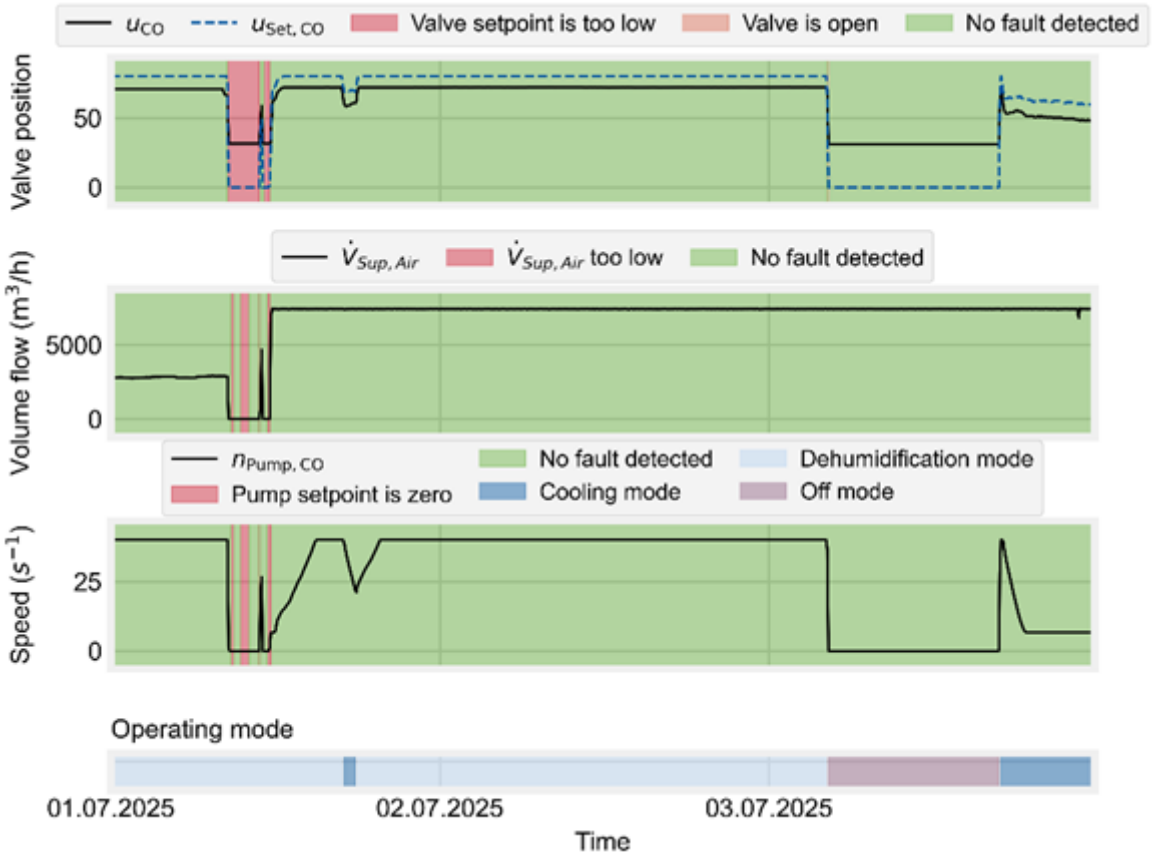

Figure 58: Fault detection and diagnosis results and operation mode for the fault types "Valve setpoint is too low", "V˙_Sup,Air too low", and "Pump setpoint is zero" for the time period from 01.07.2025 to 03.07.2025

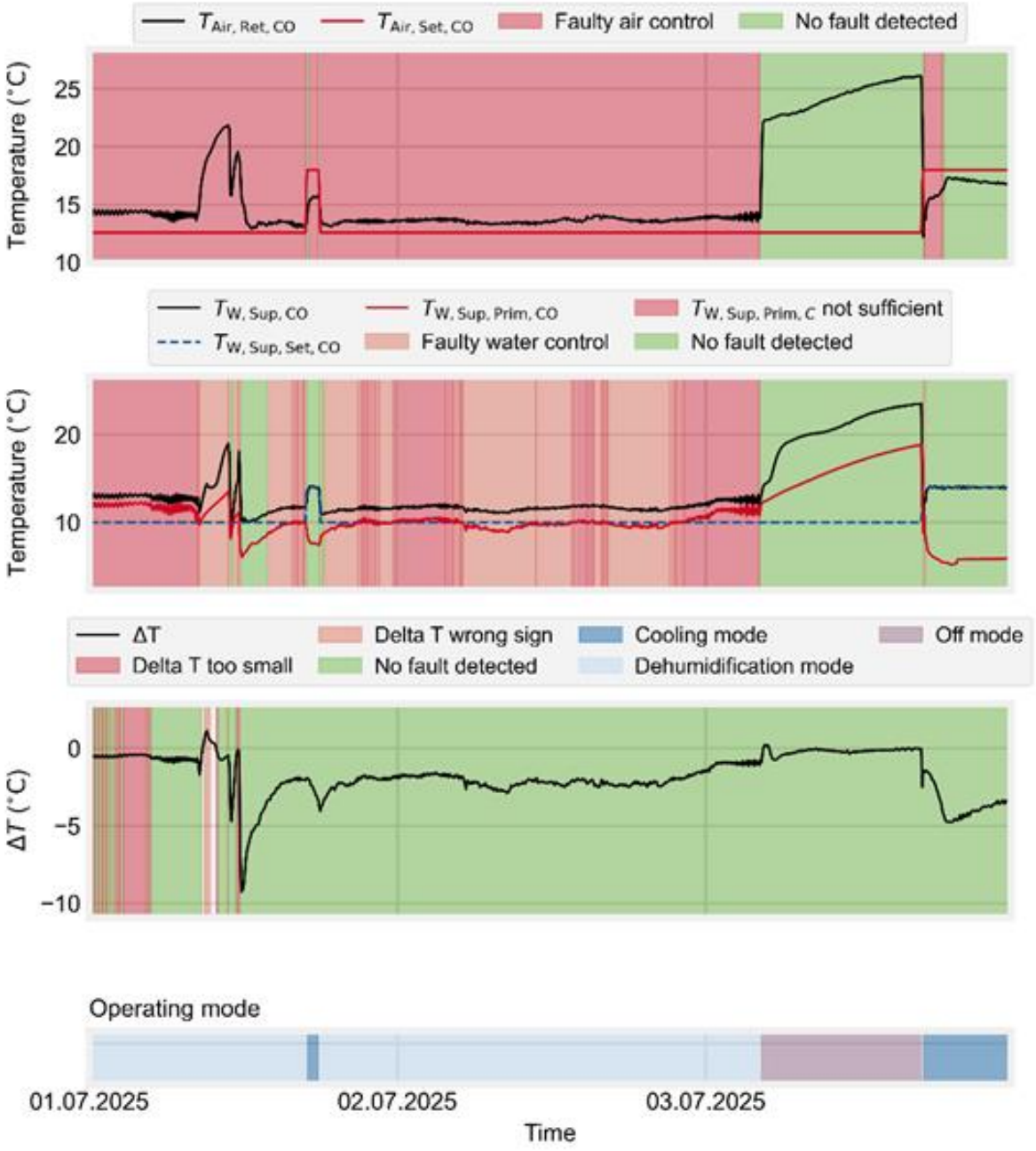

Figure 59: Fault detection and diagnosis results and operation mode for the fault types "Faulty air control", "Faulty water control", "T_W,Sup,Prim,C not sufficient" and "Delta T too small" for the time period from 01.07.2025 to 03.07.2025.

5.2.3.3 Results for the Reheater

Figure 60 displays a heatmap of the reheater's fault detection results and operating modes. Three different fault types or hierarchical fault type combinations are apparent. The fault type *valve is open* appears recurrently. *Faulty air control* is detected often together with *faulty water control*, thus both faults are analyzed together. Between the 21.10.2025 and the 22.10.2025 nearly all fault types shown in the heatmap are detected. This period will also be analyzed in detail below.

First, the fault type *valve is open* is analyzed. In Figure 61, the fault detection results, the operating mode, the valve setpoint, and the actual value of the valve are shown. The results demonstrate that the fault is detected for single time steps during the transition between the operational modes. This behavior is a byproduct of the five-minute data aggregation

window. Specifically, the mean valve position may remain non-zero despite the system having already transitioned to the off mode. To mitigate these false positives, the diagnostic logic should be refined to require a multi-step persistence criterion. Implementing such a temporal threshold would increase the methodology's robustness by filtering out transient anomalies that do not materially impact the thermal delivery of the AHU.

Figure 60: Heatmap of all fault types of the reheater occurred from 01.10.2025 to 31.12.2025 and operating mode of the reheater. To facilitate visualization, data is aggregated into 1-hour bins.

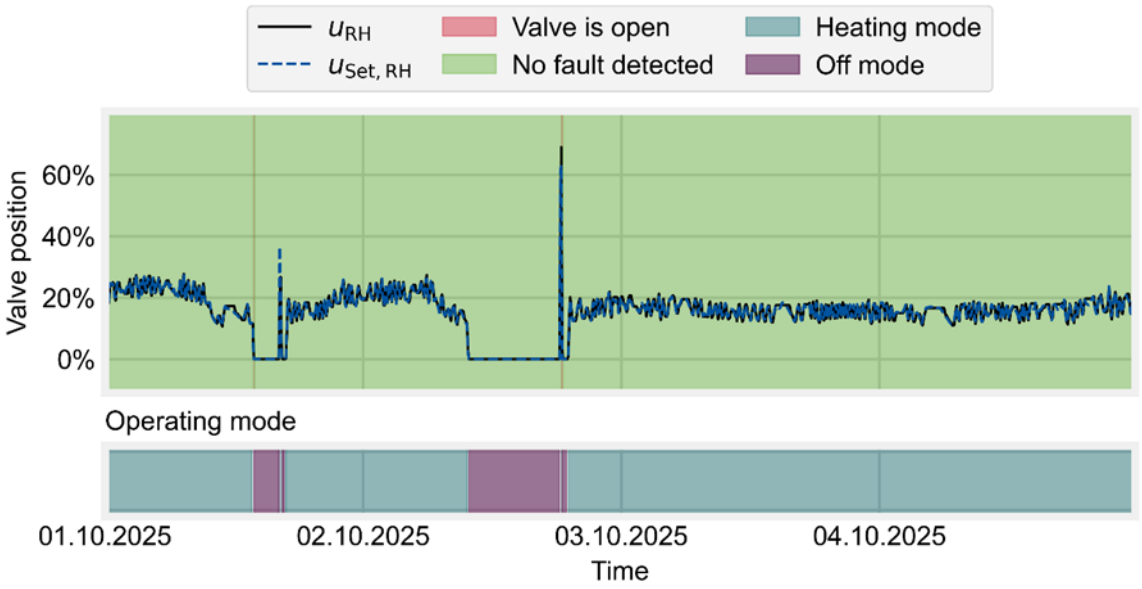

Figure 61: Fault detection and diagnosis results and operation mode for the fault type "Valve open" for the time period from 01.10.2025 to 04.10.2025.

In Figure 62, the fault detection results, operational mode of the reheater, and relevant measurements are shown from 13.11.2025 to 15.11.2025 to exemplarily analyze the fault types *faulty air control* and *faulty water control.* In this case, the fault *faulty air control* is detected when the supply air temperature is above the setpoint and the operating mode is in heating mode. The pump is already operating with its minimal allowed speed during the heating mode. This is the reason why the temperature overshoots its setpoint. The pump is only fully off during the off mode, but the switch from heating to off mode is only completed

if the extract air temperature from the room is 0.75 K above the setpoint. Thus, the overshoot is detected as a fault during the heating mode. As it is usually the goal to reach the setpoint, to solve this problem, the control strategy of the reheater must be adapted. This could be solved either by allowing a lower minimum speed of the pump or by switching earlier into off mode. In general, the fault detected in this case is a real fault as it reveals unnecessary heating of the air above its setpoint, which is inefficient. Also, the fault *faulty water control* is detected. The actual temperature oscillates around the setpoint, which could be resolved by a better parameter tuning of the PID controller.

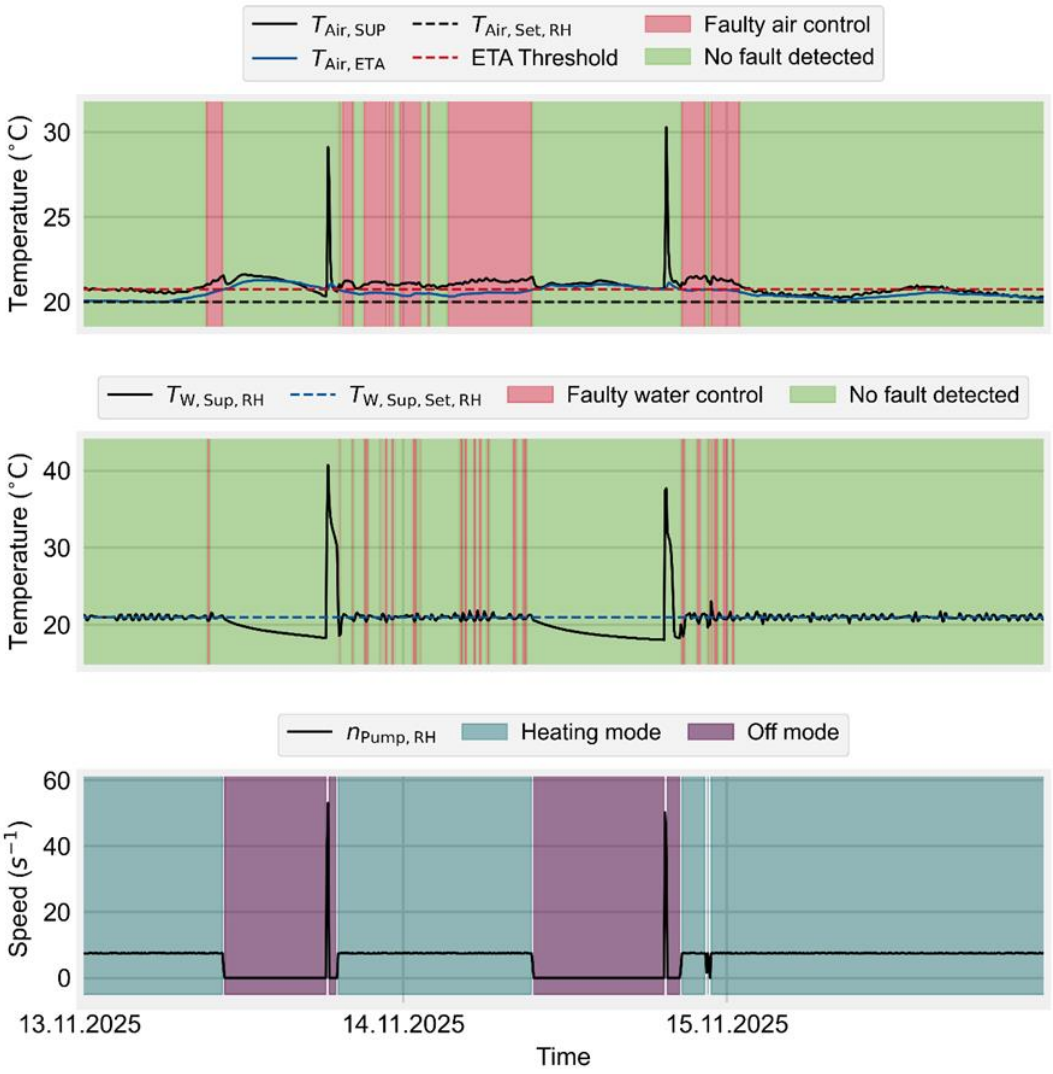

Figure 62: Fault detection and diagnosis results and operation mode for the fault types "Faulty air control" and "Faulty water control" for the time period from 13.11.2025 to

During the faulty period between 21.10.2025 and 22.10.2025 the data reveals that there must have been a communication error between the supervisory PLC and the manufacturer's PLC. The fault detection results, relevant measurements, and operating modes are given in Figure 63 for the period from 20.10.2025 to 23.10.2025. Some data points, including the air-side supply temperature, were not logged in the faulty period, which is reflected as zero values in the plot. The pump, which directly communicates with the supervisory control via BACnet, still operates and sends the actual speed back to the supervisory PLC. During this exact period the preheater is also faulty (cf.), which indicates a system wide communication error.

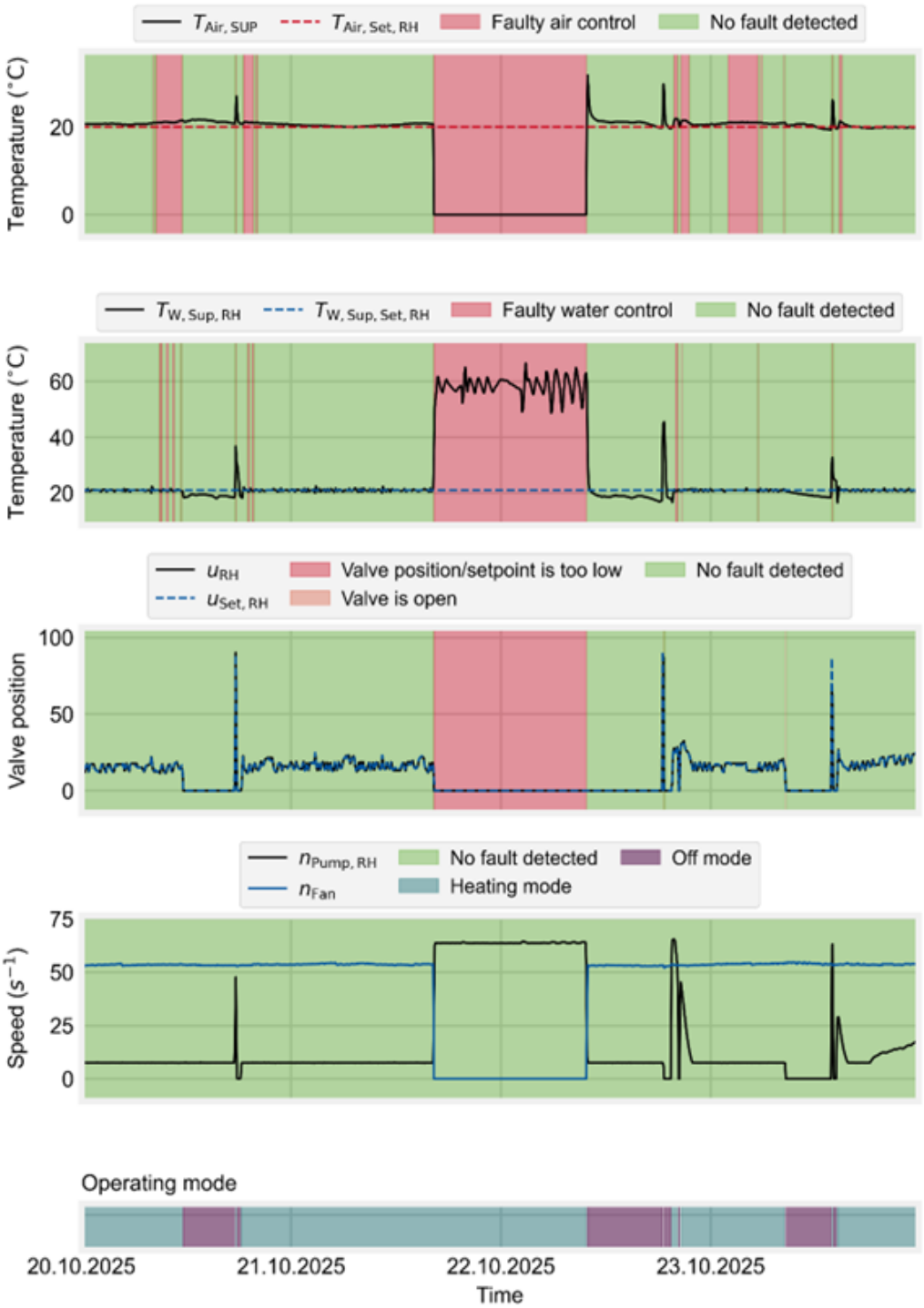

Figure 63: Fault detection and diagnosis results and operation mode for the fault types "Faulty air control", "Faulty water control", and "Valve position/setpoint too low" for the time period from 20.10.2025 to 23.10.2025.

The assessment of the FDD framework for water-to-air heat exchangers yields positive results, validating its ability to detect and isolate operational faults. The methodology consistently detects critical operational deviations, including system-level malfunctions, sensor miscalibration, and suboptimal performance. These findings demonstrate that the framework is robust and provides targeted diagnostic guidance that effectively facilitates predictive maintenance and continuous monitoring of AHU components.

5.2.4 Conclusion

The proposed rule-based FDD methodology has proven effective in uncovering operational inefficiencies, sensor miscalibrations, and equipment malfunctions, successfully validating the proof of concept for the water-to-air heat exchangers. The method demonstrates significant potential to enhance condition monitoring and support maintenance workflows for AHU components. The hierarchical diagnostic structure successfully isolates initial fault indicators based on available sensor measurements. However, it is important to note that these indicators often represent the earliest detectable symptoms rather than the fundamental physical root causes. Consequently, definitive diagnosis in many cases still requires human expert intervention. Some false alarms are still prevalent, which could be avoided by a multi-step persistence criterion. Future developments will focus on expanding this methodology to cover all AHU components. The developed methodology contributes to the reliable and efficient operation of buildings, effectively transforming them into predictable components of an energy community. This consistency is essential for collective energy management, as it minimizes unforeseen demand spikes, secures grid stability, and maximizes the overall energy flexibility required to meet local decarbonization targets.

5.3 Multi-Layered HVAC Systems Harmonization Scheme (TU Wien)

To bridge the gap between unstructured building data and domain-specific engineering standards, we developed and implemented a multi-layered data harmonization framework (Figure 64). This pipeline ingests heterogeneous data sources and transforms them into a unified, topology-aware context optimized for downstream tasks such as LLM-based fault detection and explanation generation.

5.3.1 Heterogeneous Data Ingestion Layer (Data Sources)

The primary challenge in modern building automation is the vast disparity in data formats, which this layer systematically normalizes upon ingestion to prepare the system for downstream LLM application. It simultaneously imports geometric and semantic asset data from Industry Foundation Classes (`IFC / BIM Files`), structured relational metadata using the `Brick Schema`, and domain-specific fault definitions through `AFDDOnto`. Concurrently, the layer handles high-frequency, dynamic time-series telemetry delivered via `BMS Sensor CSV` files directly from physical building equipment. By capturing spatial, semantic, and telemetry streams at a single point of entry, this layer establishes the raw ingredients necessary to build a comprehensive digital twin, creating the structured data foundation required before an LLM can be deployed on top of the system to interpret HVAC behaviors.

5.3.2 Semantic Extraction and Ontology Harmonization Layer (Modules 1 & 2)

Once ingested, the raw and disconnected data streams are passed to the semantic extraction pipeline to establish a unified vocabulary, which is essential if an LLM is to later reason over the system without misinterpreting sensor definitions. The `IFCIngestor` extracts physical asset properties and maps them into the `OntologyStore`, while the `BMSLoader` and `FeatureEngineer` transform raw sensor points into physically meaningful thermodynamic attributes, such as temperature differentials or pressure drops. The core harmonization occurs within the `SemanticMapper` and `OntologyAligner`, which resolve naming discrepancies and bind the loose time-series telemetry streams to their exact physical counterparts within the structural ontology. Finally, the `KGBuilder` compiles these verified relationships into an enriched Knowledge Graph (KG) validated by the `KGEvaluator`, outputting highly structured metadata files including `components.json` and `connections.json` that establish the explicit relational truth on top of which an LLM can reliably operate.

Figure 64: Data Sources Integration

5.3.3 Topology-Aware Semantic Retrieval Layer (Module 3)

To enable a Large Language Models (LLMs) to be effectively utilized on top of this architecture, the harmonized metadata must be indexed into a format that a generative agent can query without losing critical spatial and functional relationships. The `ChunkBuilder` partitions the unified component data into semantic text fragments, which are subsequently converted into dense vector embeddings using a `sentence-transformers/miniLM-L6-v2` model and stored within a `FAISS Vector Store`. When a query is initiated, the `QueryGuard` filters the input for security and relevance, allowing the system to execute a semantic lookup that is instantly cross-referenced with the explicit `TopologyGraph`. The `PromptBuilder` then merges these vector search results with the precise connectivity maps of the fluid and airflow loops, synthesizing a context-rich prompt payload that preserves thermodynamic dependencies, making it ready for an LLM to read and analyze.

5.3.4 LLM Orchestration and Fault Explanation Agent Layer (Module 4)

This layer represents the reasoning tier that sits directly on top of the harmonized data structure, showing how an LLM can be deployed to translate dense, graph-enriched engineering context into natural, diagnostic narratives for human operators. Incoming requests are processed through a localized `QueryGuard` and paired with the topology-aware context before being submitted to the core `LLM Backend` for generation. To ensure that the

generative model remains bounded by engineering realities, the architecture routes the raw textual output through a `ResponseParser` and a specialized `HallucinationChecker`. This verification pipeline is designed to cross-check the LLM's diagnostic assertions against the active sensor anomalies and physical boundaries established in the underlying knowledge graph, demonstrating how an LLM agent can safely deliver a `Structured Fault Explanation` when layered on top of this framework.

5.3.5 Continuous Multi-Stage Evaluation Layer (Module 5)

To ensure the operational reliability required for industrial building deployment, the architecture incorporates a comprehensive, multi-stage evaluation scheme designed to validate the system at every stage of the pipeline. This framework establishes the necessary methodology to isolate and benchmark performance across specialized modules: the `KGEvaluator (M1)` for semantic graph topology precision, the `BMSEvaluator (M2)` for feature engineering mathematical integrity, the `RAGEvaluator (M3)` for context retrieval relevancy, and the `LLMEvaluator (M4)` to score the final text generation. These independent metrics are structured to aggregate into an automated `eval_results.json` log. This evaluation layer provides the definitive validation blueprint required to flag regressions in data alignment, context fetching, or language generation, ensuring a reliable benchmark framework is in place when the final diagnostics are rendered via the Flask interface.

6 Technical Workshops (TU Wien, Chalmers)

The technical workshops were organized to share knowledge and foster collaboration. These workshops focused mainly on the challenges and techniques in Machine Learning & Physics-Informed AI and Synthetic Data Generation.

7 References

[1] C. Siegl, G. Schweiger, R. Kraus, T. Mach, M. Moerth, und T. Hirsch, „Data and Sourcecode from: Load Prediction for Mixed Use Districts“. DiLT Analytics, 2024. doi: 10.5281/zenodo.13329857.

[2] T. Schranz, Q. Alfalouji, T. Hirsch, und G. Schweiger, „An open IoT platform: Lessons learned from a district energy system“, in *2022 second international conference on sustainable mobility applications, renewables and technology (SMART)*, IEEE, 2022, S. 1–9.

[3] J. Heaton, *Introduction to Neural Networks for Java*, 2. Aufl. St. Louis: Heaton Research, Inc, 2008.

[4] C. Bergmeir und J. M. Benítez, „On the use of cross-validation for time series predictor evaluation“, *Information Sciences*, Bd. 191, S. 192–213, 2012, doi: 10.1016/J.INS.2011.12.028.

[5] R. J. Hyndman und G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[6] N. R. Draper und H. Smith, *Applied regression analysis*, Bd. 326. John Wiley & Sons, 1998.

[7] IEA. “World Energy Outlook 2025 – Analysis,” November 12, 2025. https://www.iea.org/reports/world-energy-outlook-2025/

[8] E. Crowe, Y. Chen, H. Reeve, D. Yuill, A. Ebrahimifakhar, Y. Chen, L. Troup, A. Smith, and J. Granderson. Empirical analysis of the prevalence of HVAC faults in commercial buildings. In: Science and Technology for the Built Environment 29.10 (Nov. 2023), pp. 1027–1038. ISSN: 2374-4731. DOI: 10.1080/23744731.2023.2263324.

[9] Y. Li and Z. O'Neill. An innovative fault impact analysis framework for enhancing building operations. In: Energy and Buildings 199 (Sept. 2019), pp. 311–331. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2019.07.011.

[10] J. P. Teichmann, J. Oppermann, P. Matthes, D. Müller, P. Mathis, and A. Kümpel. Reducing energy consumption of air handling units by optimized pump control. de. Tech. rep. RWTH-2018-224864. Conference Name: Roomvent&Ventilation; 2018. SIY Indoor Air Information Oy, 2018.

[11] M.H. Schraven, M. Baranski, A. Kümpel and D. Müller 2023. A comprehensive building HVAC design for application of model-predictive control: Experiences and challenges of construction, commissioning, and operation in a real-world scenario. Universitätsbibliothek der RWTH Aachen. https://doi.org/10.18154/RWTH-2023-03811