ECom4Future
Fiware Driven ENERGY COMmunities for the FUTURE

DELIVERABLE

3.1 Preliminary Combined Report for Energy Analytic

Version 1.1

Adil Mukhtar, TU Wien

Gerald Schweiger, TU Wien

David Steen, Chalmers

Elena Malakhatka, Chalmers

Radu Plamanescu, UPB

Irene Hafner, DWH

Christof Sumereder, FH Joanneum

Theresa Kohl, DiLT Analytics

Christoph Siegl, DiLT Analytics

Marwa Maghnie, RWTH Aachen University

Internal Reference

Deliverable No.:	D 3.1 (2024)
Deliverable Name:	**Preliminary 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 report offers a concise summary of the technical approaches discussed in the literature for synthetic data generation, forecasting, and fault detection and diagnosis (FDD). It highlights the current progress and activities carried out this year. A variety of methods for generating synthetic data, including model-based, data-driven, and AI-based techniques for forecasting, load profiling and fault detection and diagnosis (physics), are commonly used and adapted for measurement devices. This work package task will concentrate on the techniques and analysis that align most closely with the goals and synergistic aims outlined 1by ECom4Future.

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, 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.

In summary, this report reviews techniques for generating synthetic electricity consumption and production profiles, emphasizing model-based and AI-driven approaches. It covers the use of stochastic models, such as those found in RichardsonPy, to create realistic power profiles by setting parameters based on real-world data. Additionally, the report examines how AI-based methods can capture complex data correlations and dependencies using parametric distributions. A preliminary analysis of electricity consumption profiles from a real energy community and its participants is included, along with the development of initial forecasting models for hourly electricity consumption. The report also presents a framework for fault detection and diagnosis specific to air handling units. A novel FDD framework for this purpose has been developed, tested, and published at a conference (DX24).

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.

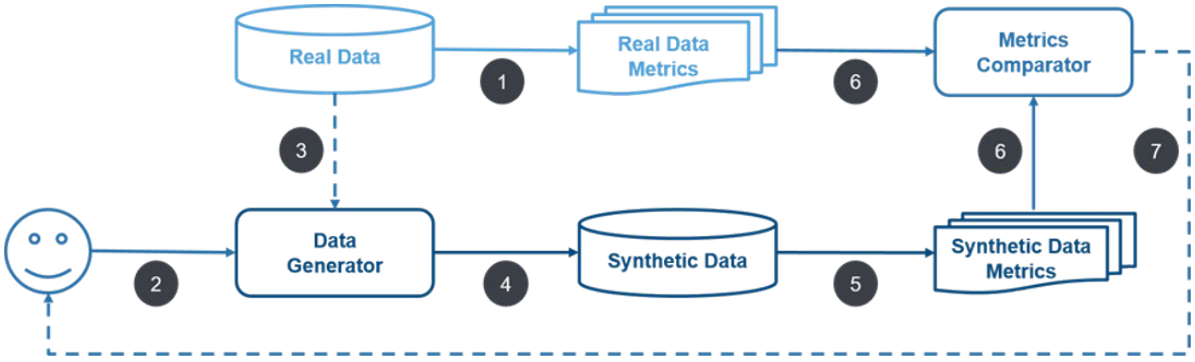

Figure 1: Synthetic data process

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.

2.2.1 Holistic anomaly detection (RWTH)

Detecting and diagnosing faults in air handling units (AHUs) requires data on system operations, both fault-free and faulted. However, real-world datasets with labeled faults are often scarce. Synthetic data offers a practical solution to simulate operational scenarios for fault detection algorithms.

Requirements:

- High temporal resolution: Fault dynamics in AHUs require data sampled at intervals of 1 minute or 15 minutes.
- Annotated fault scenarios: Simulations should include different fault types and intensities to cover a wide operational spectrum.
- Digital twins: Accurate modeling of components, such as coolers and valves, is essential for generating realistic synthetic data.

Parametric Scope

- Variables: Air and water temperature, humidity, flow rates.
- Timeframes: Short-term faults (hours) and long-term behaviors (weeks).
- Tools: Modelica and EnergyPlus for physics-based simulations

Use Case:

- Holistic anomaly detection within an air handling unit (AHU)
- Measurement Devices: Sensors for air and water temperatures, humidity etc.
- Resolution: 1 minute, 15 minutes
- Digital Twin: Coolers, and other AHU devices
- Annotations: Likelihood of scenarios, Faulty scenarios

2.2.2 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.

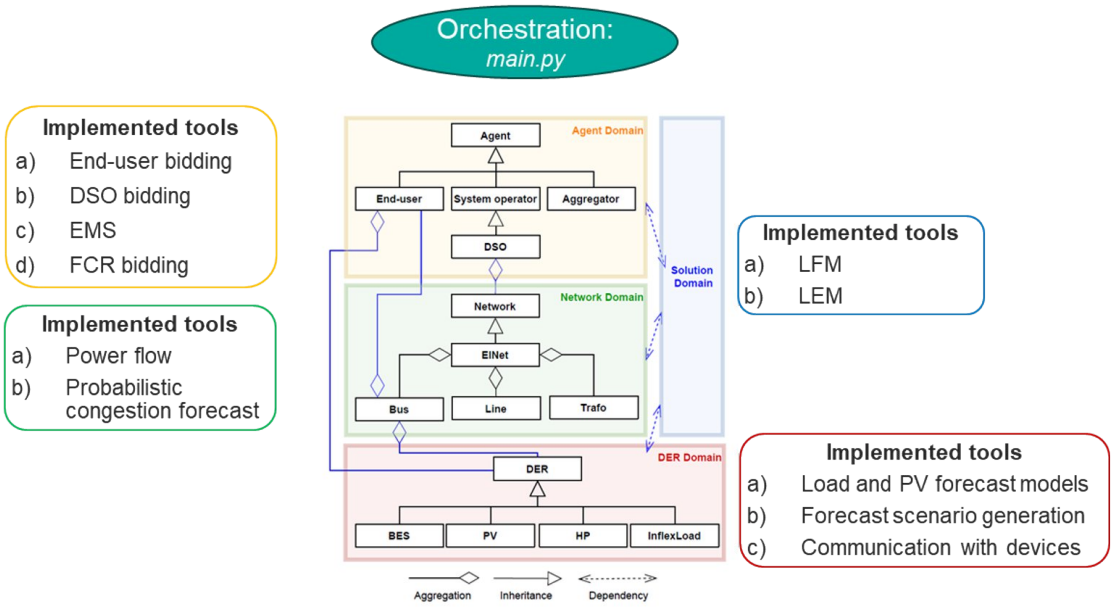

Figure 2: Local Flexibility market ABM (Chalmers use case)

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

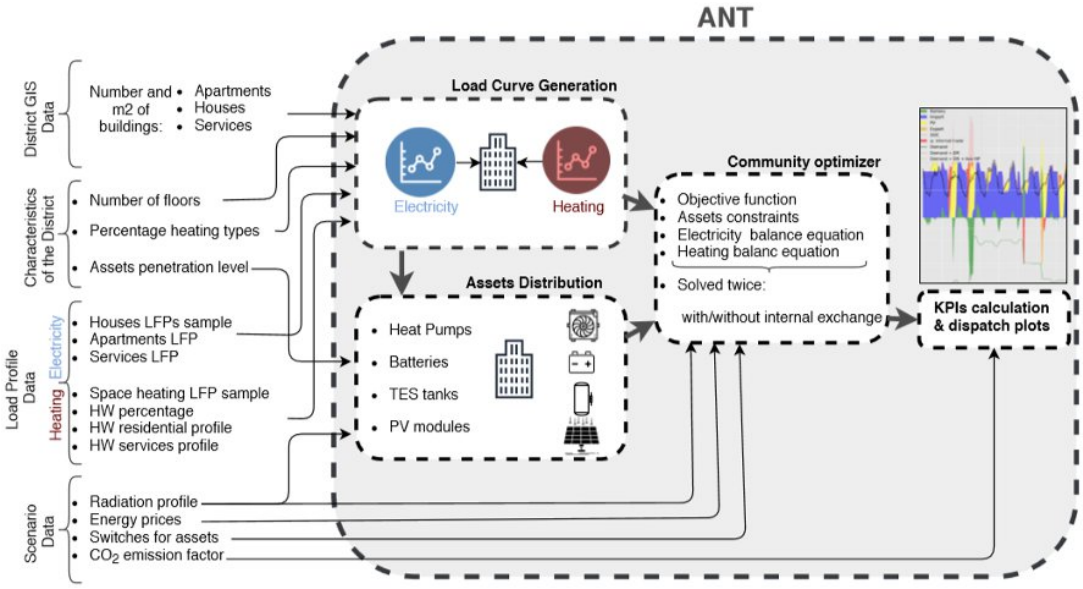

Figure 3: ANT model (Chalmers use case)

2.2.3 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

2.2.4 Physics-Informed AI Fault Detection and Diagnosis (TUWien)

Physics-informed AI models provide a robust framework for fault detection in systems like photovoltaic (PV) arrays, batteries, and EV chargers by integrating physical system constraints into the learning process.

Requirements

- High-frequency data: Synthetic data at resolutions of 1 minute to hourly intervals.
- Comprehensive fault scenarios: Include faults across components such as inverters and storage systems.
- Physical realism: Maintain adherence to operational constraints and dynamics.

Parametric Scope

- Variables: Power output, fault intensity, state of charge (SOC).
- Tools: Variational Autoencoders (VAEs) for generating high-fidelity fault scenarios.
- Digital twins: PV arrays and battery storage systems

Use Cases

- Fault Detection and Diagnosis
- Measurement Devices: Appliances, PV, Batteries
- Resolution: 1 minute, 15 minutes, Hourly
- Digital Twin: PV, Batteries, Charging Stations, etc.
- Annotations: Faulty scenarios, Fault intensity

2.2.5 Data analytics for improved Forecasting Load and Generation Profiles (UPB)

2.2.5.1 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).

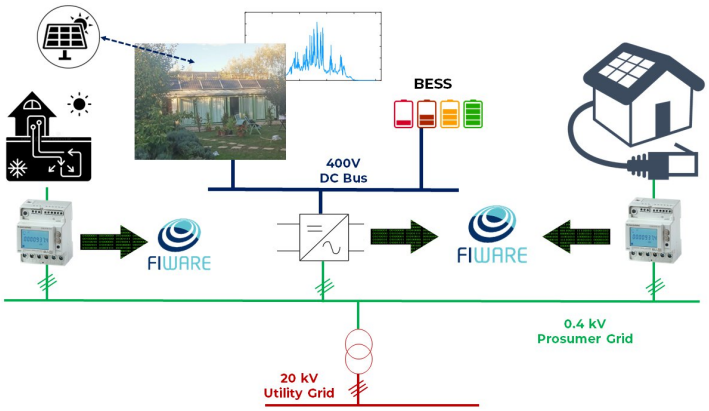

Figure 4: GreenMogo Framework

2.2.6 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.

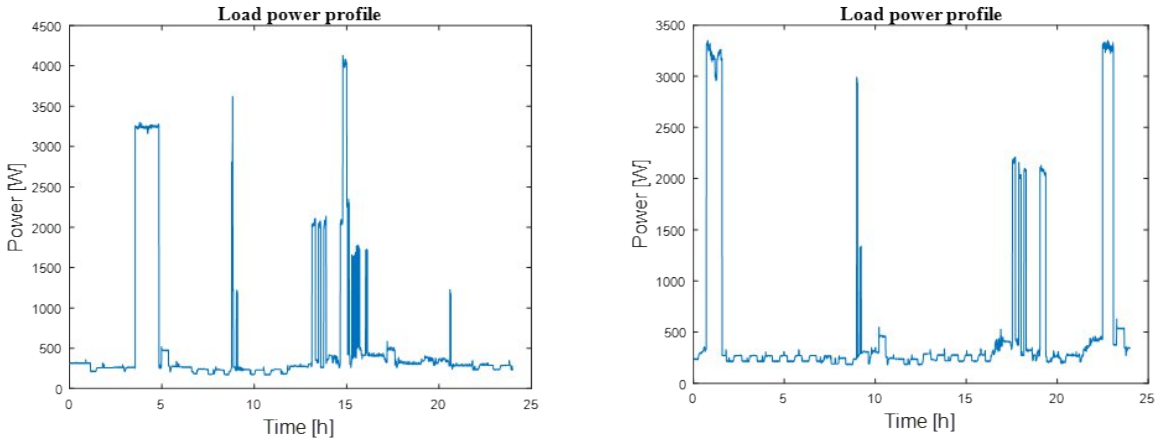

Figure 5: Power Profiles

2.2.6.1 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.

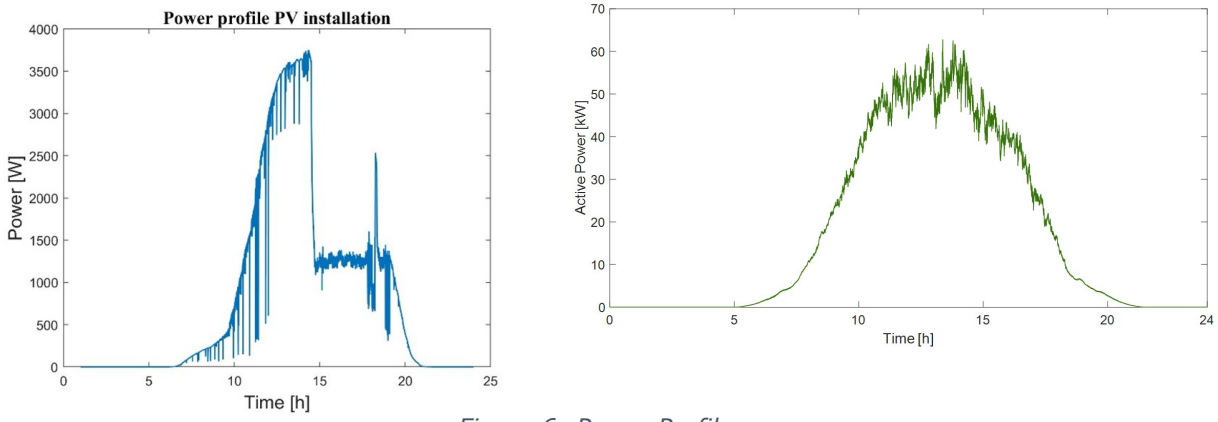

Figure 6: Power Profiles

2.2.6.2 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, $\bar{y}$ – 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; $\tilde{y}$ – 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].

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

where $\bar{y}_p$ 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 $\bar{y}_p$ is also the mean $\bar{y}$ of the model samples y_i calculated on T_r; the choice of representative model value $\bar{y}_p$ 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.

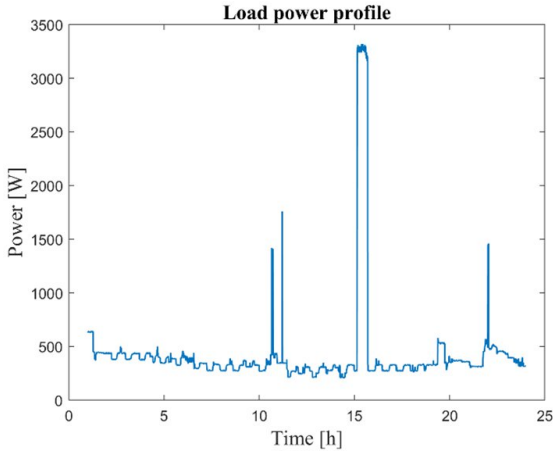

Figure 7: Power Profile

Figure below presents the metric CV(RMSD) applied to load power profiles for a daily analysis window (24 hours) and hourly constant profile model (T_a = 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 T_{r16}. In addition, the Figure below presents the metric CV(RMSD) applied to load power profiles for an analysis window T_a = 24 hours and T_r = 30 minutes. One can observe that the maximum CVRMSD is 130% depicted at 3:00 pm, corresponding to T_{r30}. More over, the third picture below presents the metric CV(RMSD) applied to load power profiles for an analysis window T_a = 24 hours and T_r = 15 minutes. One can observe that the maximum value is 121 % depicted at 3:30 pm, corresponding to T_{r62}

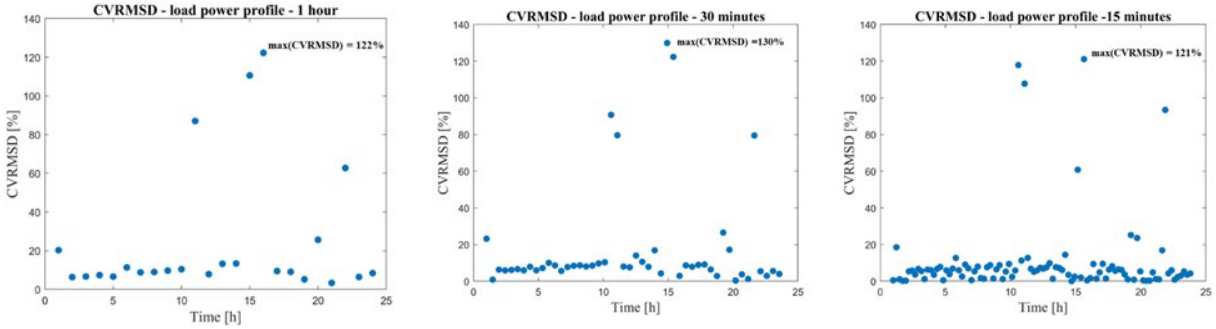

Figure 8: CV (RMSD) Analysis

2.2.6.3 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 $P_{max} = 8\,kw$ 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.

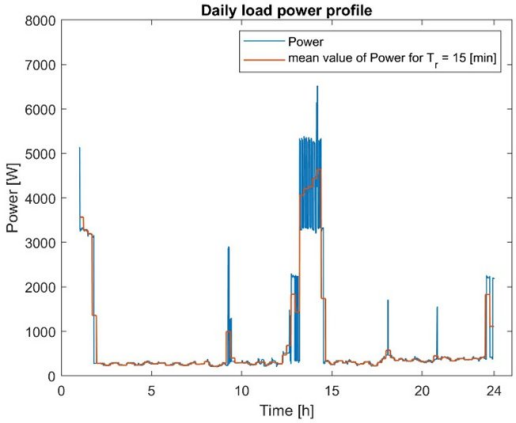

Figure 9: Power Profile

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

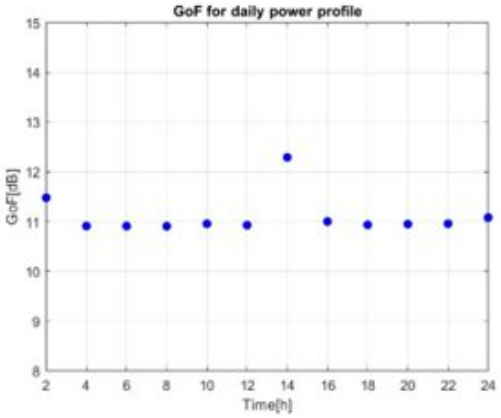

Figure 10: GoF Analysis

We repeat the procedure for one month (March 2023) to calculate daily GoF results using an analysis window T_r = 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:

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

where $\tilde{P}_i$ 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.

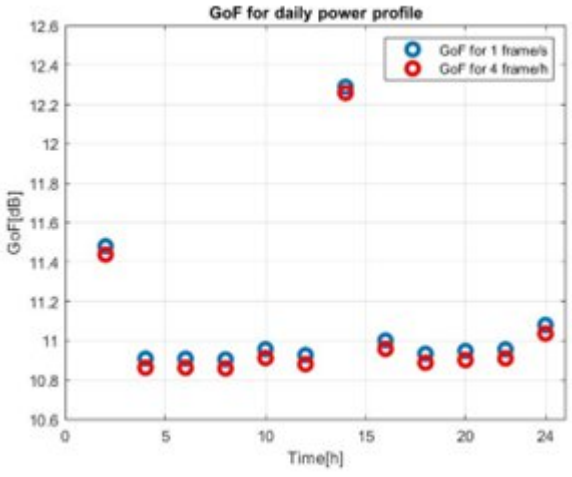

Figure 11:GoF Analysis

2.2.6.4 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

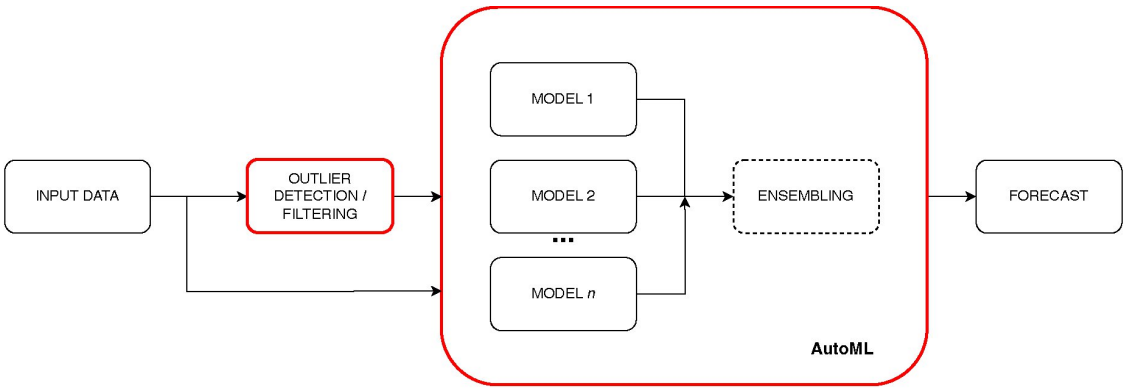

Figure 12: Anomaly Detection Process Flow

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.

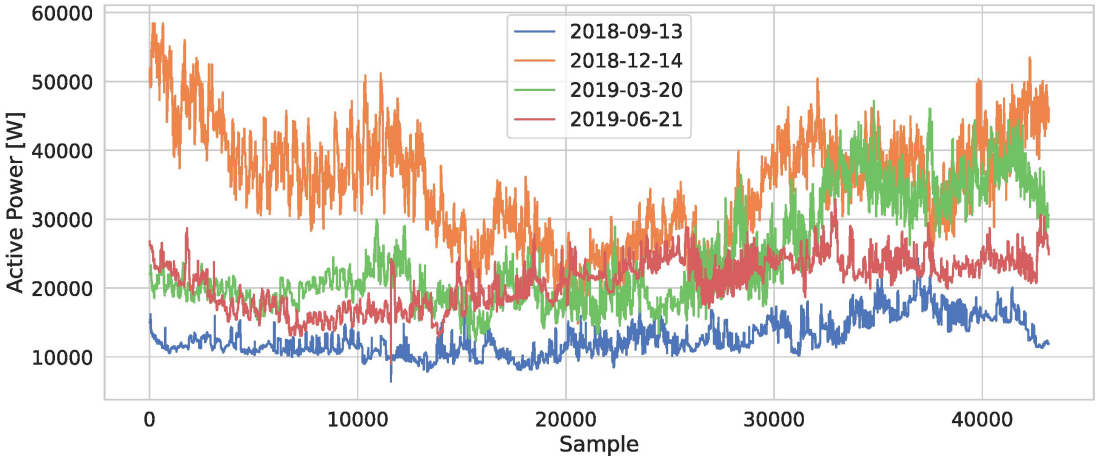

Figure 13: Politechnica Bucharest, Romania Campus Trends Overview

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.

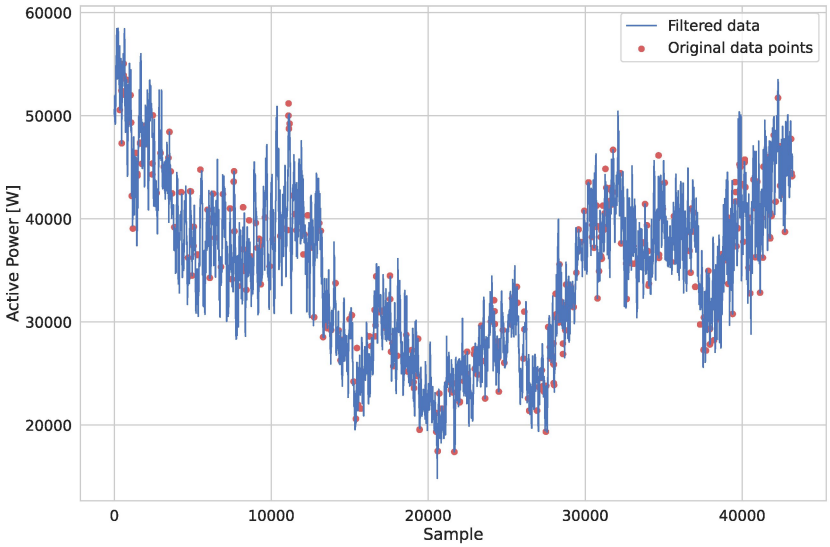

Figure 14: Hampel Filter Analysis for Outlier Detection

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.

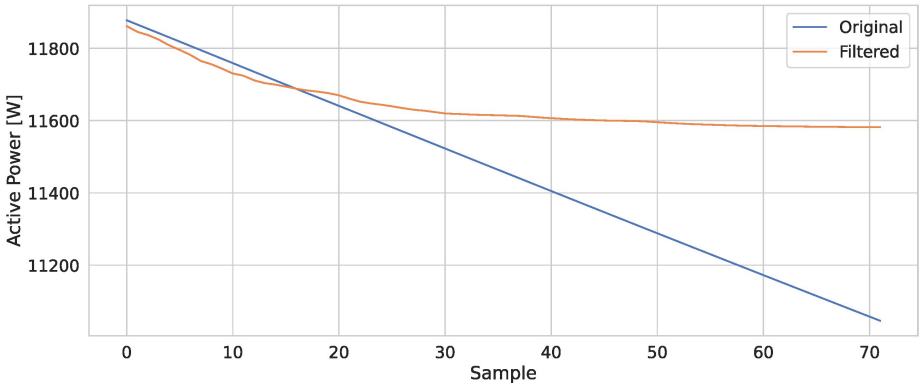

Figure 15: Prediction Outcomes

3 Austria-Mariatrost Energy Community Preliminary Electricity Consumption Analysis and Forecasting (TU Wien, DiLT, dwh)

The Renewable Energy Community (EEG) in Mariatrost 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 Mariatrost 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.

Figure 16: Mariatrost Energy Community (Link)

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.

3.1 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.

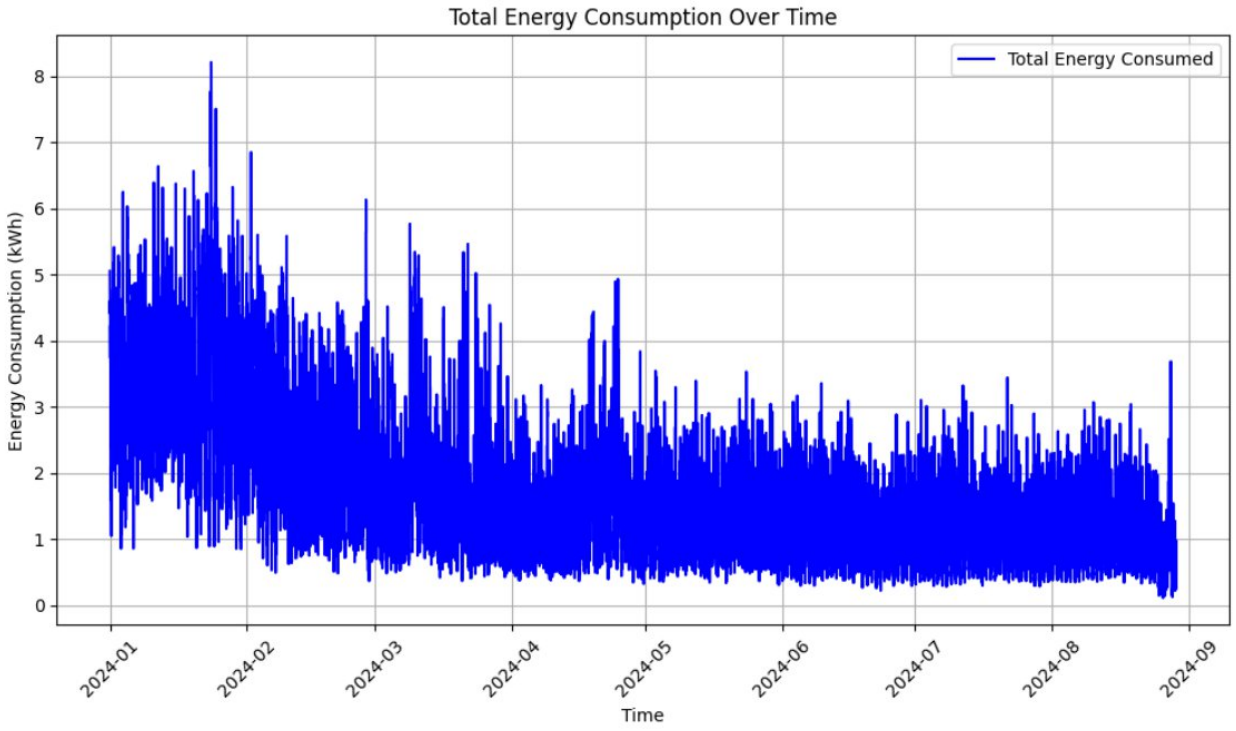

Figure 17: Total Energy Consumption (Aggregated 15-mins) of EC Mariatrost

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.

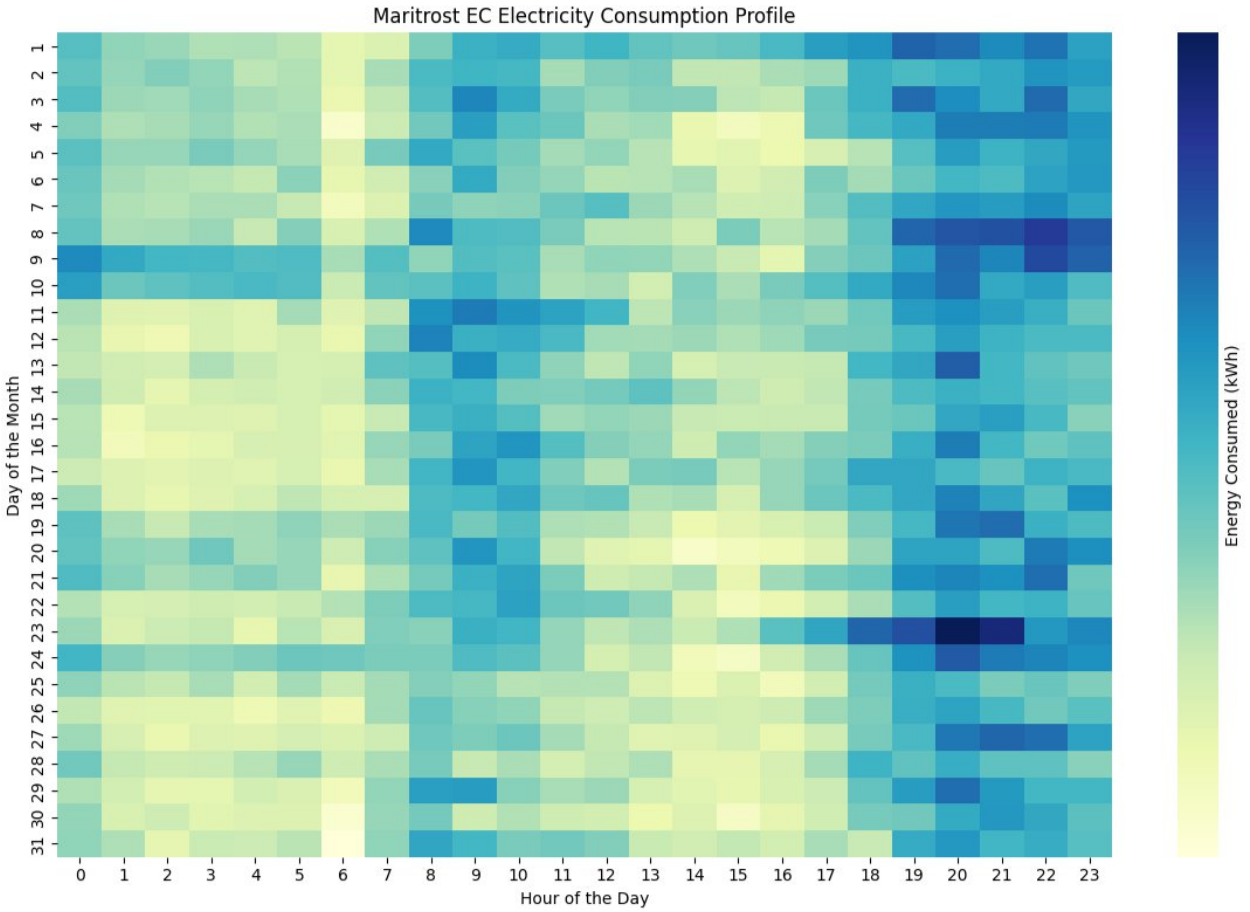

Figure 18: Electricity Consumption Patterns During Days of the Month

3.2 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.

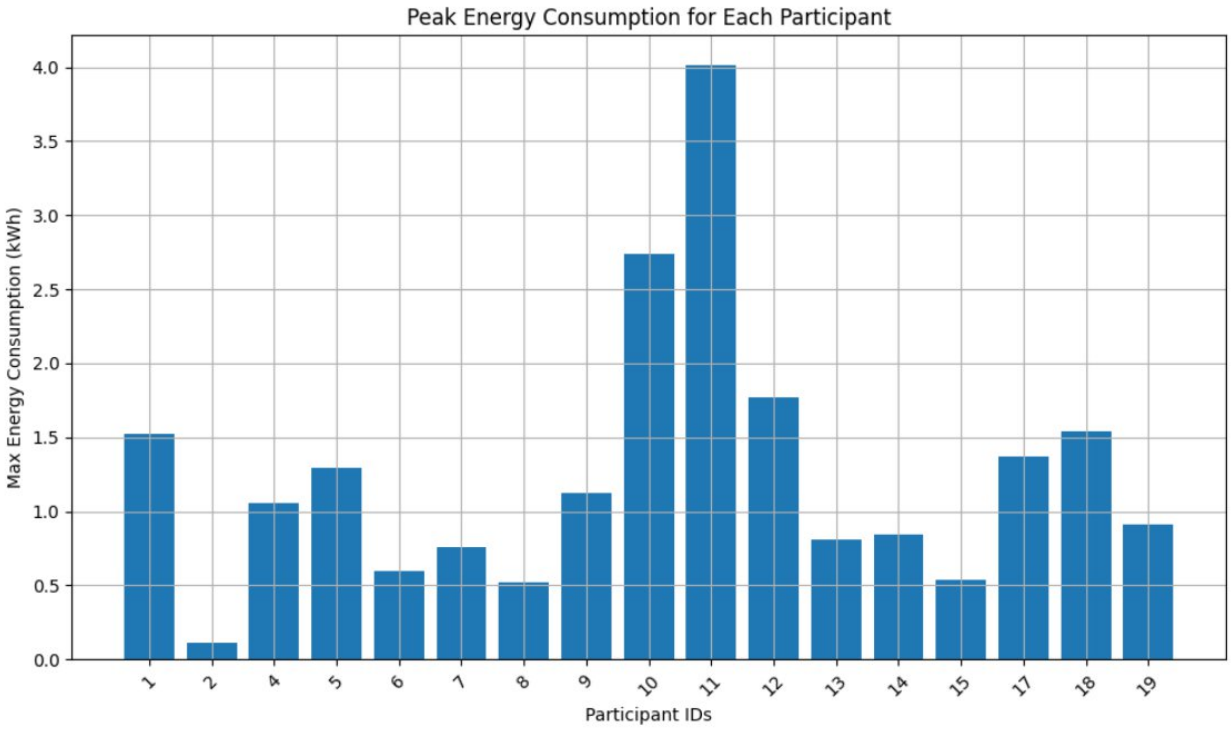

Figure 19: Max Energy Consumption for Each Participant

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.

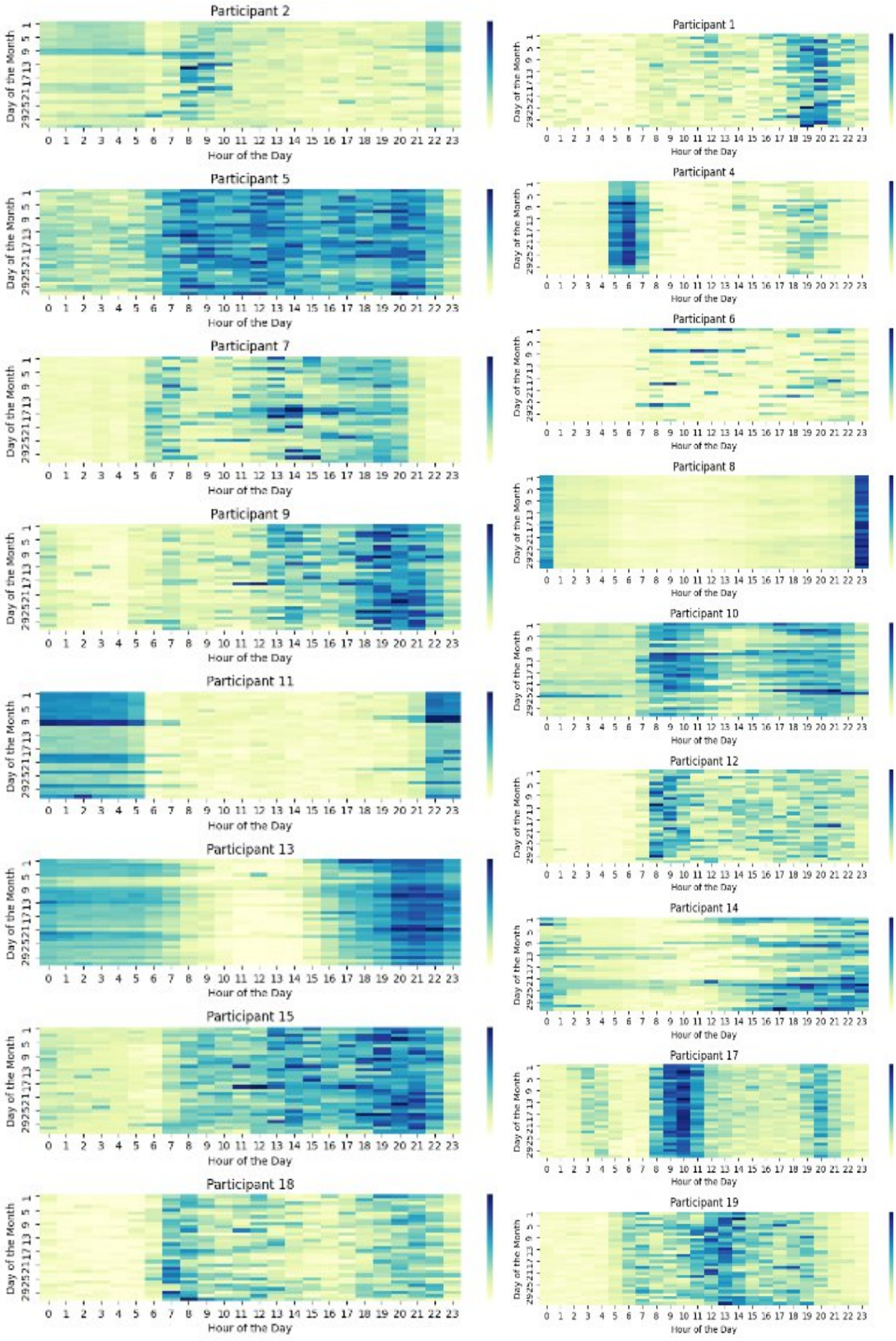
Participant 2
Participant 1
Participant 5
Participant 4
Participant 7
Participant 6
Participant 9
Participant 8
Participant 11
Participant 10
Participant 13
Participant 12
Participant 15
Participant 14
Participant 18
Participant 17
Participant 19
Day of the Month
Hour of the Day

Figure 20: Participants Electricity Consumption Profile

3.3 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.

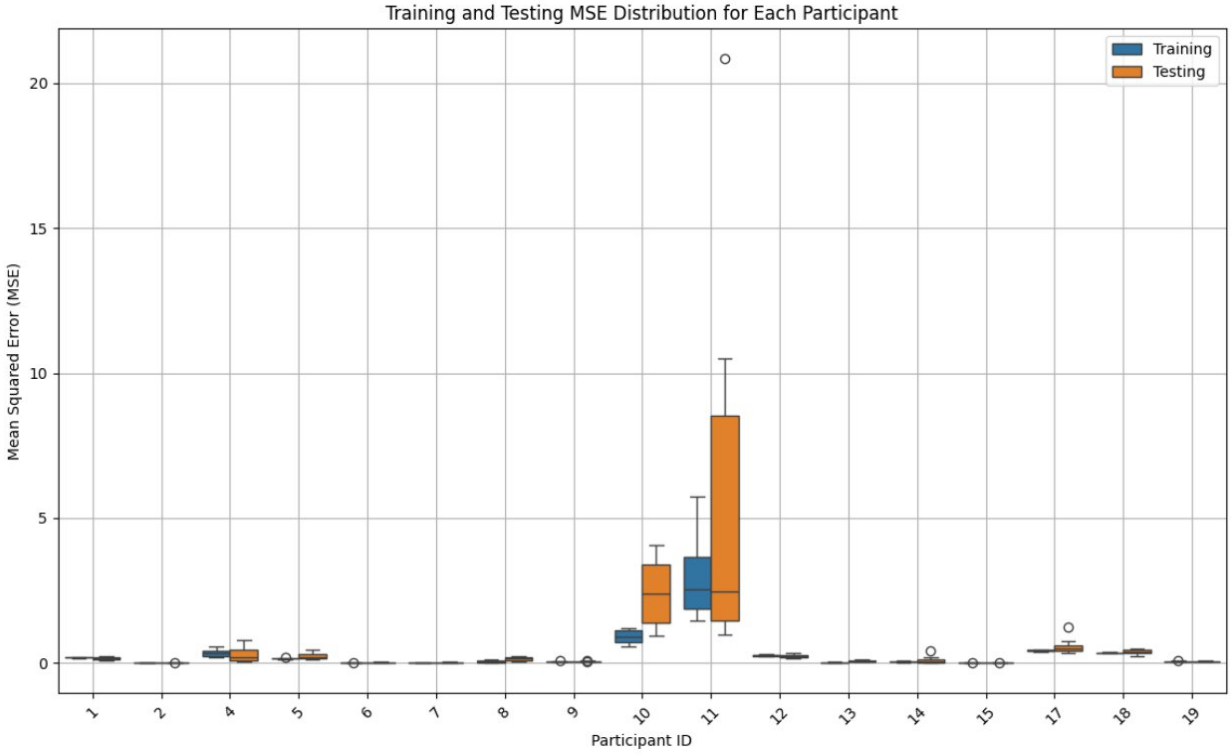

Figure 21: Average MSE for Baseline Model for Hourly Energy Demand Prediction

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 Mariatrost 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 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 22). 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.

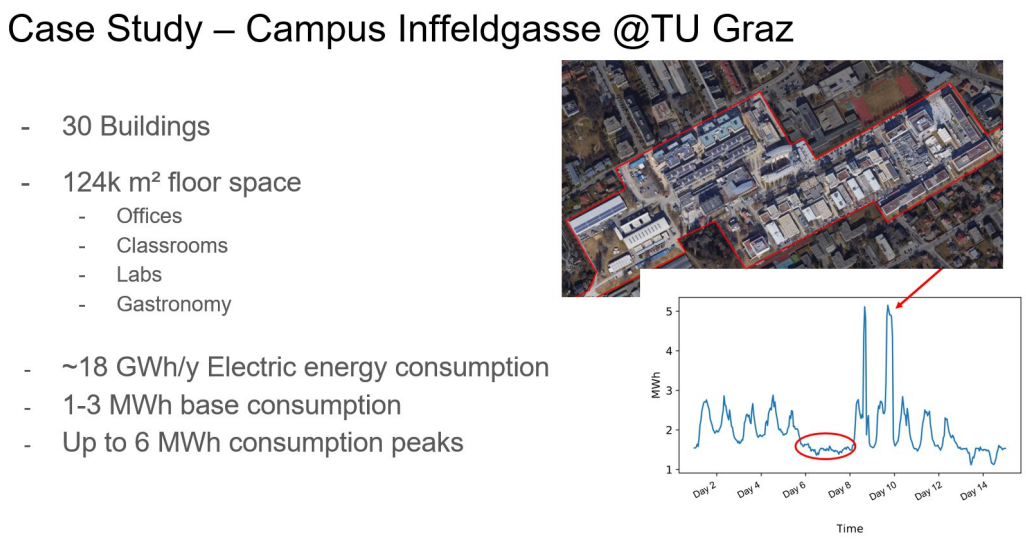

Figure 22: Key facts about the Case Study

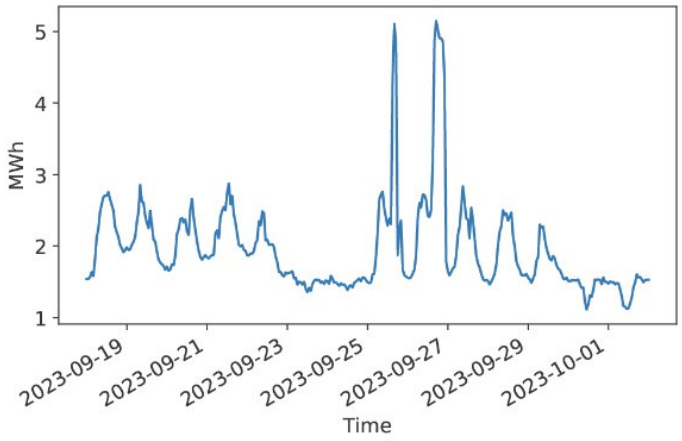

Figure 23. The total energy consumption of the Inffeld campus over two weeks.

Figure 22 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.

4.1 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].

4.2 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[1] (v1.4.0) and XGBoost[2] (v2.0.3) libraries. The neural network models are implemented using Keras[3] (v2.10.0) and TensorFlow[4] (v2.10.0), and the applied model architecture and size are based on existing literature and established rules of thumb [2].

4.3 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 $F_1..F_n$ and the performance of the model is then measured on fold F_{n+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.

4.4 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.

[1] https://scikit-learn.org
[2] https://xgboost.readthedocs.io
[3] https://keras.io/
[4] https://www.tensorflow.org/

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 24 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.

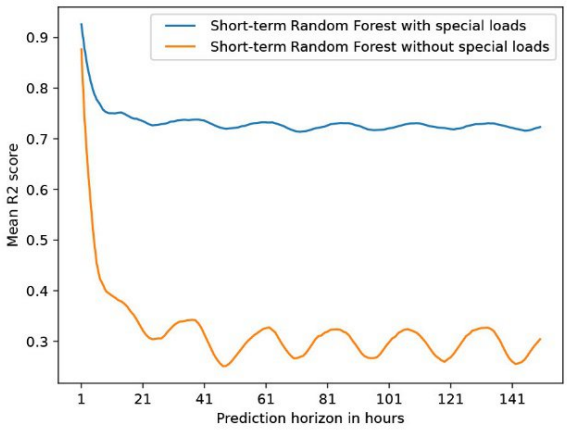

Figure 24. Short-term models based on Random Forest with and without special loads as features over a prediction horizon of 150 hours.

Long-Term Prediction Models Performance:
Table 3 shows the cross-validation results of the long-term energy prediction models utilizing special load scheduling information. As the used models do not rely on loads for previous time steps, the reported scores are the result of averaging the prediction performance of all points in time within the cross-validation test splits (test split sizes ranging in size from 88 to 112 days). Again, Random Forest models perform best at the task of load prediction in all metrics (e.g., *R2=0.80*). The Bayesian Ridge model achieves the second-best performance in terms of MSE and R2 (*MSE=145*, *R2=0.79*).

Table 3. Mean scores of long-term predictions after time-series cross-validation.

Model	MSE (MWh)	R2	MAE (kWh)	MAPE
XGBoost	161	0.75	250	0.11
Linear Regression	149	0.78	263	0.12
Elastic Net	407	0.41	396	0.18
Decision Tree	151	0.76	246	0.11
Random Forest	**119**	**0.80**	**220**	**0.10**

Bayesian Ridge	145	0.79	257	0.12
ANN	157	0.78	251	0.11
DNN	177	0.75	305	0.11

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 25 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.

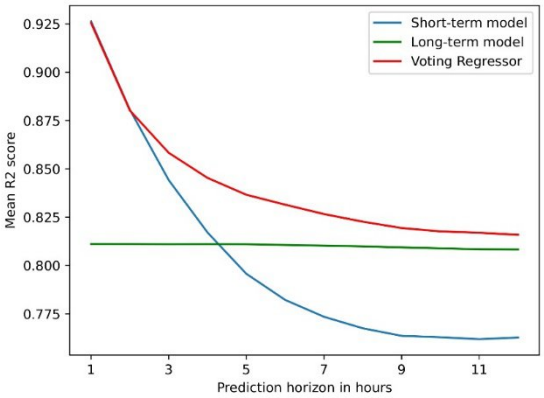

Figure 25: Comparison of the best performing short- and long-term models (both Random Forests).

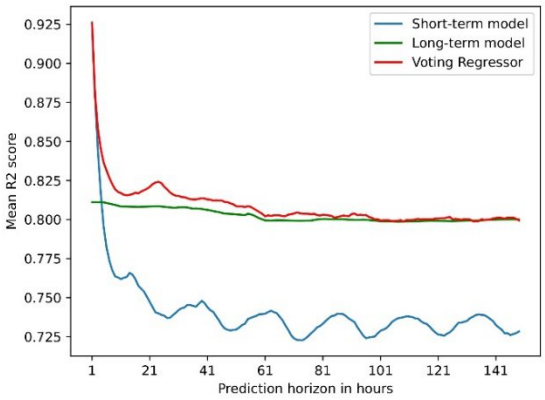

Figure 26: R2 scores from cross-validation of the Voting Regressor, short- and long-term models over a prediction horizon of 150 hours.

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 25 and 26 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 27 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.

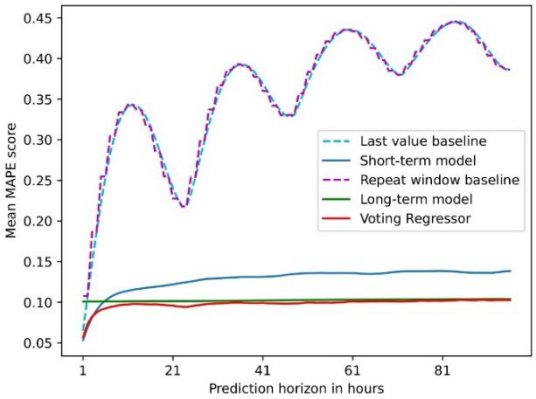

Figure 27: MAPE scores from cross-validation of

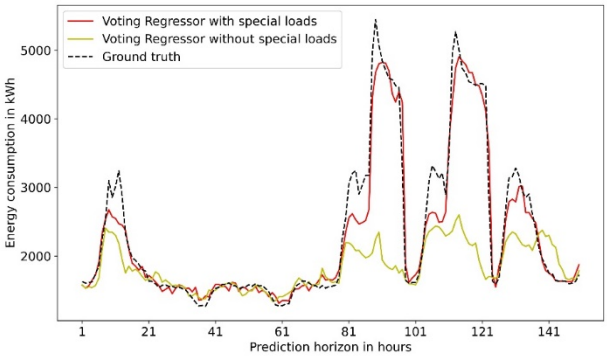

Figure 28: Energy prediction using the Voting

the Voting Regressor, short- and long-term models and the trivial baselines over a prediction horizon of 96 hours.

Regressor with and without lab schedule information for the next 150 hours.

Figure 28 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 29 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).

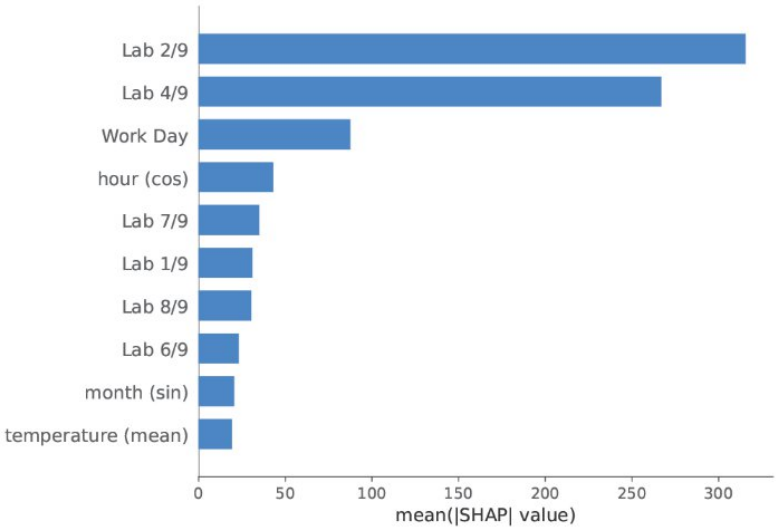

Figure 29. Top ten features of SHAP analysis of Random Forest performing long-term prediction.

4.5 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 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 Motivation & 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.

5.2 Framework and Methodology

The overall framework of OCCEN is presented in Figure 30. 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.

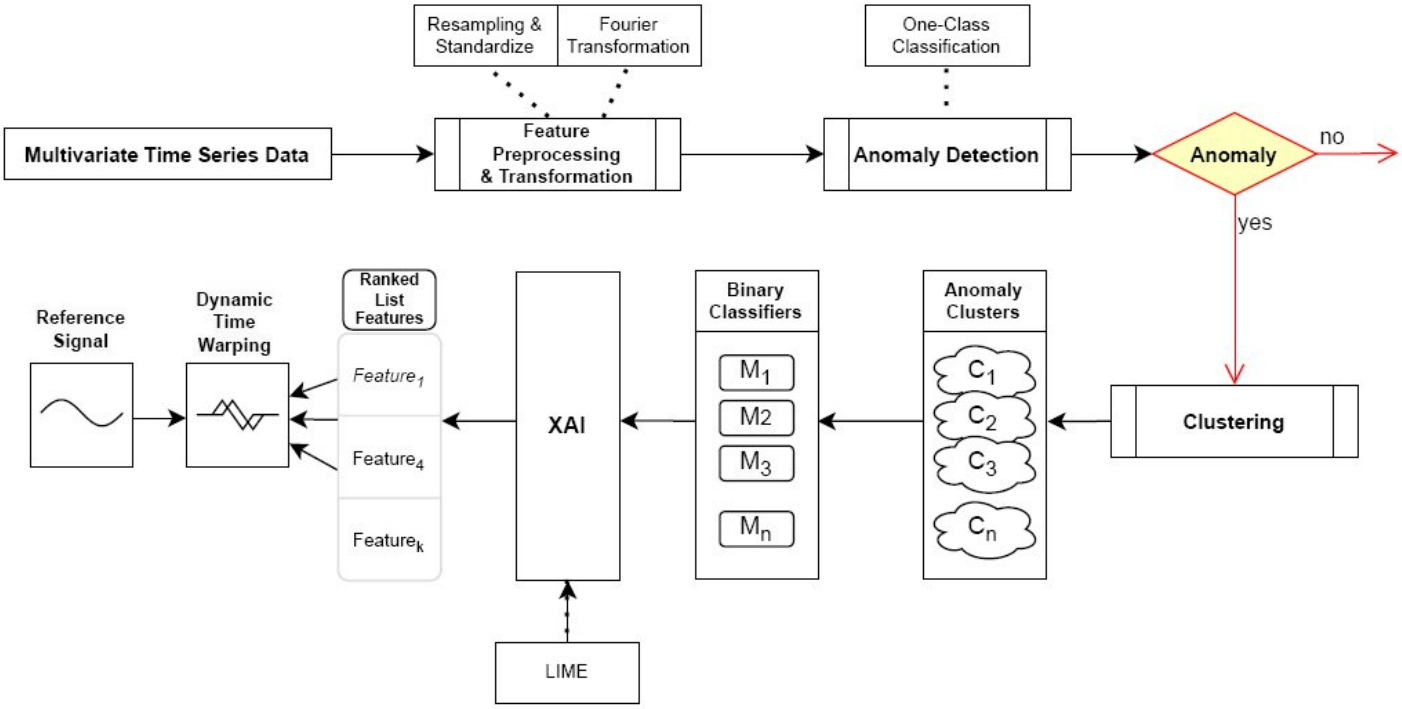

Figure 30: Anomaly Detection and Diagnosis Framework

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 31.

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.

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 31: Diagnosis through Anomaly Detection, Clustering, and Feature Ranking

5.3 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 32), 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 32: Summary of Fault Imposition Methods for AHU

5.4 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 33. 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.

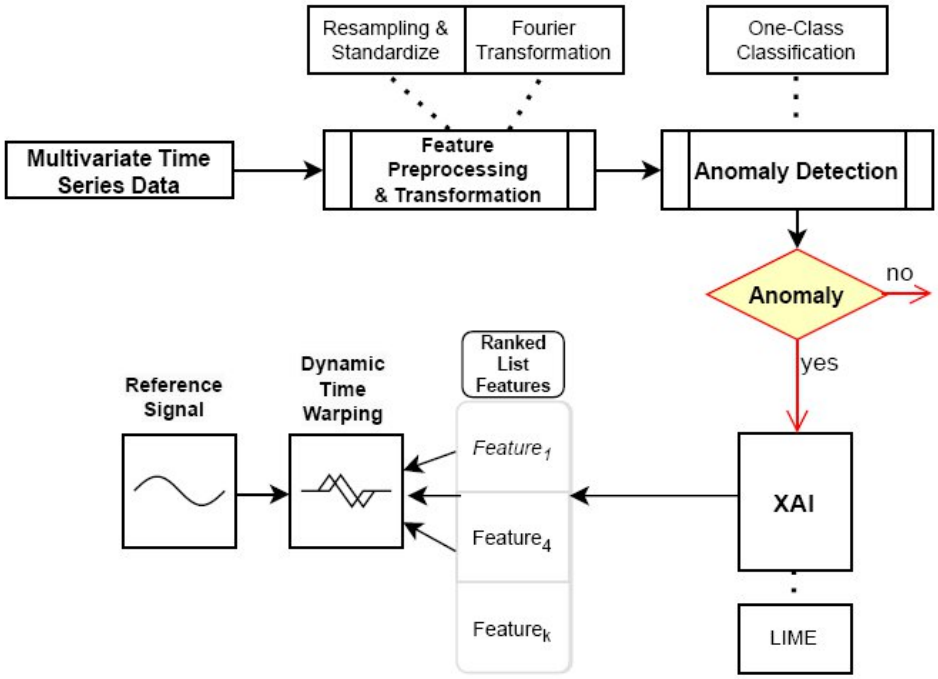

Figure 33: Baseline Approach Design

5.5 Results

5.5.1 Fault Detection

The average results of the anomaly detection task are presented in Figure 34. 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.

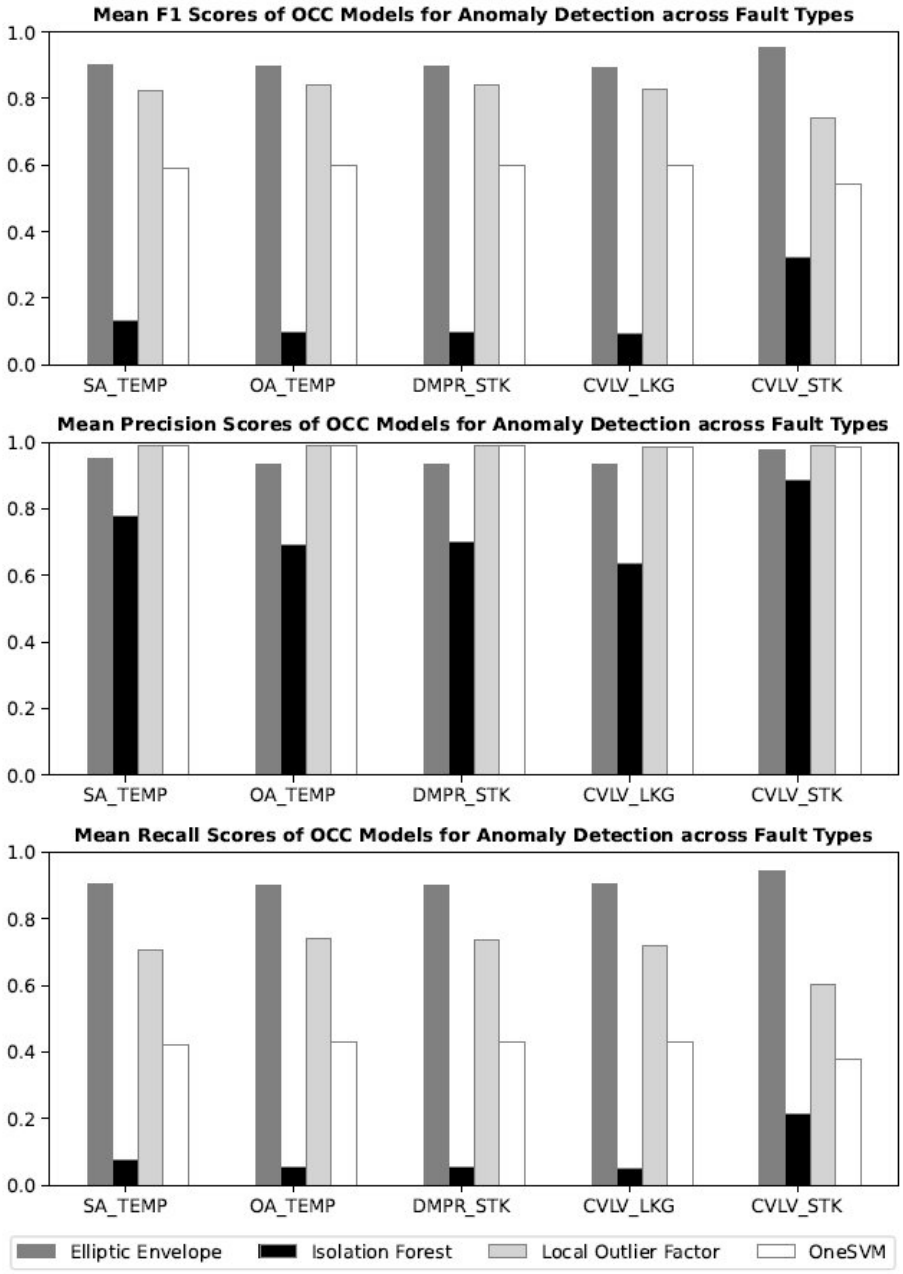

Figure 34: 10-Fold Cross Validation Evaluation

5.5.2 Fault Diagnosis

Overlap@P: The baseline and OCCEN comparative analysis is presented in Figure 35. 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.

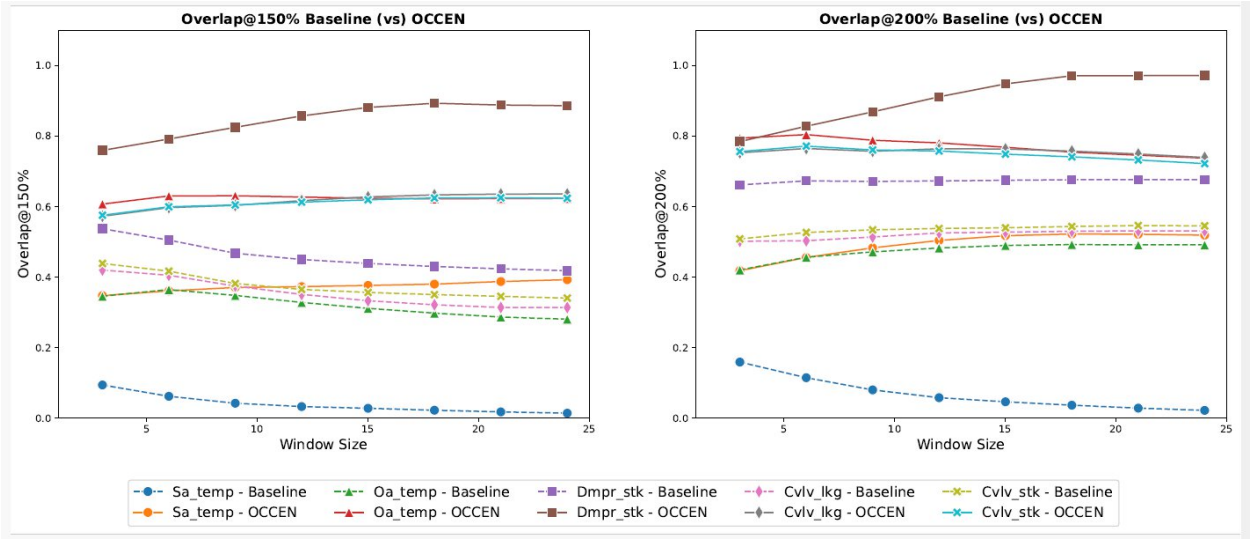

Figure 35: Mean score comparison of Overlap@P metric for baseline and OCCEN

HitRate@K: The overall evaluation of HitRate@k metric is presented in Figure 36. 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.

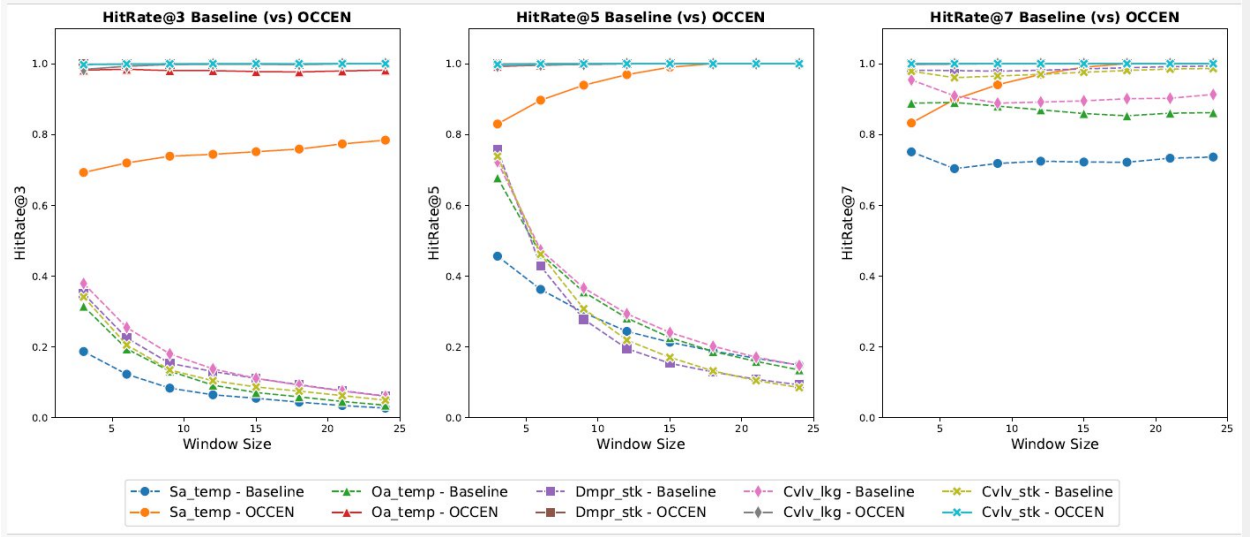

Figure 36: Mean score comparison of HitRate@k metric for baseline and OCCEN

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 37 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.

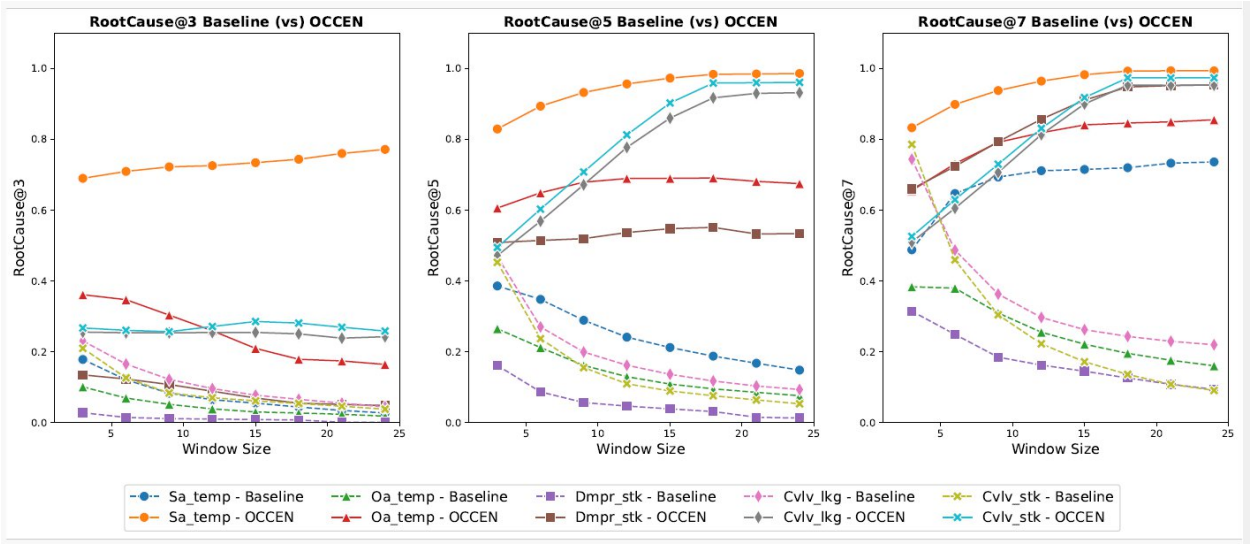

Figure 37: Mean score comparison of RootCause@k metric for baseline and OCCEN

5.6 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 already published in a conference and will be indexed in the coming year.

6 Holistic anomaly detection (RWTH)

6.1 Motivation and use case

According to the World Energy Outlook [7] report from 2024, almost 37 % of the world's electricity use is within the building sector (IEA 2024). Therefore, the RWTH trial site focuses on optimizing the energy use in an experimental hall building. The building was commissioned by the Institute for Energy Efficient Buildings and Indoor Climate of the E.ON Energy Research Center (E.ON ERC). The experimental hall (Figure 38) is separated into two areas: one where test benches and building energy system components are located and another where there is an office workspace. The climate in the experimental hall is largely controlled by an air handling unit (AHU) (Figure 39). The AHU is capable of conditioning the air within the whole building, including heating, cooling, humidification, and dehumidification. The different AHU components are visualized in Figure 40. To allow efficient building energy maintenance and optimization, the building energy system in the experimental hall is monitored via a FIWARE-based cloud platform, which is briefly described in the next section.

Figure 38: Test hall building for the RWTH use case

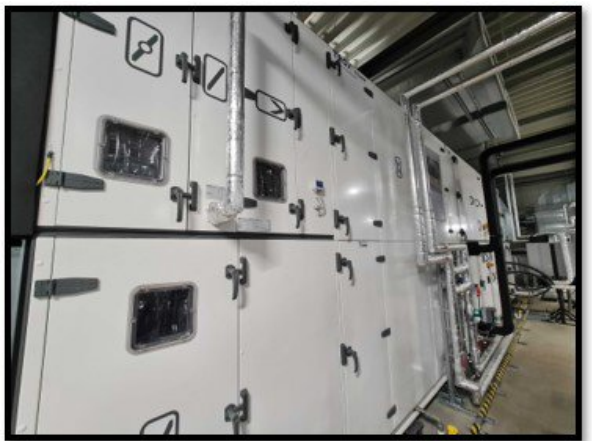

Figure 39: Air handling unit in the experimental hall

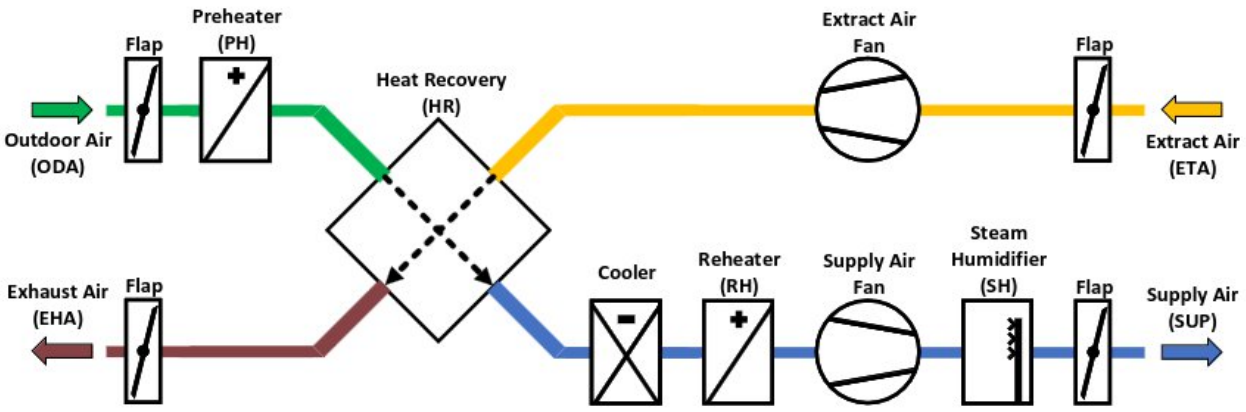

Figure 40: Schematic of the air handling unit and its components [9]

6.2 Underlying software framework

To facilitate the integration of building components, FIWARE is utilized as a central broker for communication, as shown in Figure 41. A software stack utilizing the FiLiP python library was developed for streamlined access to the FIWARE broker and integration of custom software components. All monitoring data as well as component models are stored in the cloud. The

method for creating the models is described in the following section.

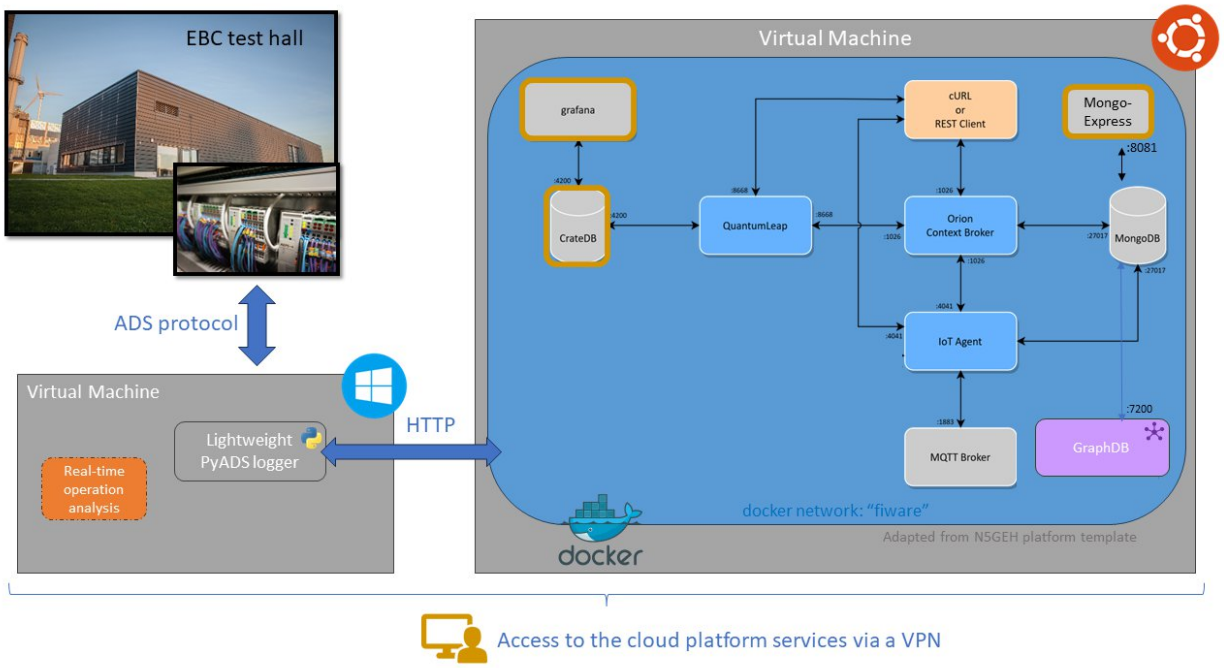

Figure 41: Cloud platform that hosts monitoring data and services for analytics

6.3 Probabilistic modeling and validation

6.3.1 Modeling approach with probabilistic graphical models

Probabilistic graphical models (PGMs) are system models with an intrinsic representation of uncertainty. The use of graph representations in PGMs aids in interpretability and inference applications. There are multiple types of PGMs, including Markov fields and Bayesian Networks (BNs). While Markov Fields are undirected, BNs are direct acyclic graphs. BNs are especially suitable for fault detection and diagnosis applications (Lu et al. 2024).

6.3.2 Data set for model training

Data from a single year is chosen to represent most possible system states regarding weather changes. As the training step requires a large number of instances to capture system dynamic state probabilities sufficiently, the data is processed at a time resolution of 15 minutes. Finally, since the training data is from a real system with faults and downtime, the data is cleaned. This entails interpolating missing data or removing whole time steps in case of downtimes with a longer duration.

For training (and subsequently validating) the models, the Python Package PySMILE from BayesFusion LLC[5] is used. PySMILE provides numerous convenience scripts for implementing probabilistic models, specifically, Bayesian Networks. The chosen method for training is Bayesian Search. The Bayesian Search structure learning algorithm is among the earliest and most widely used methods in this domain. The algorithm primarily employs a hill-climbing approach [8]. This process is guided by a scoring heuristic, which in this use case, is the log-likelihood function, and includes random restarts to enhance exploration.1

[5] https://www.bayesfusion.com/smile/

6.3.3 Evaluation metrics

To evaluate the model accuracy, the following KPIs are used:

- Mean absolute error

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|f_i - o_i|$$

where N is the number of reading predictions, f_i is the predicted value for the i-th reading instance, and o_i is the actual reading value of the i-th instance.

- Brier score loss

$$\text{Brier score} = \frac{1}{N}\sum_{i=1}^{N}(f_i - o_i)^2$$

where N is the number of predictions, f_i is the predicted probability for the i-th event, and o_i is the actual outcome of the i-th event (0 or 1). The score loss is considered in this use case, therefore, the objective should be minimized.

6.3.4 Validation results

While training the Bayesian Network models, the optimal bin size for discretizing the continuous time series was examined. The KPIs from Section 6.3.1 were plotted for different discretization granularity values, as shown in Figure 42. The MAE and RMSE drop significantly once the bin size is around 20. This shows that the granularity of the model does not have to be high in order to have good enough results. The Brier score loss on the other hand peaks around 20 bins, then also decreases steadily. Since the Brier score measures whether the predicted values was exactly in the bin with the highest probability, any values predicted to be in near-by bins would be penalized. The acceptability of this behavior is subject to tolerance of error margins per use case. For the present use case, a perfect bin match is not required. Considering these hyperparameter results, a bin size of 20 is chosen for the probabilistic models.

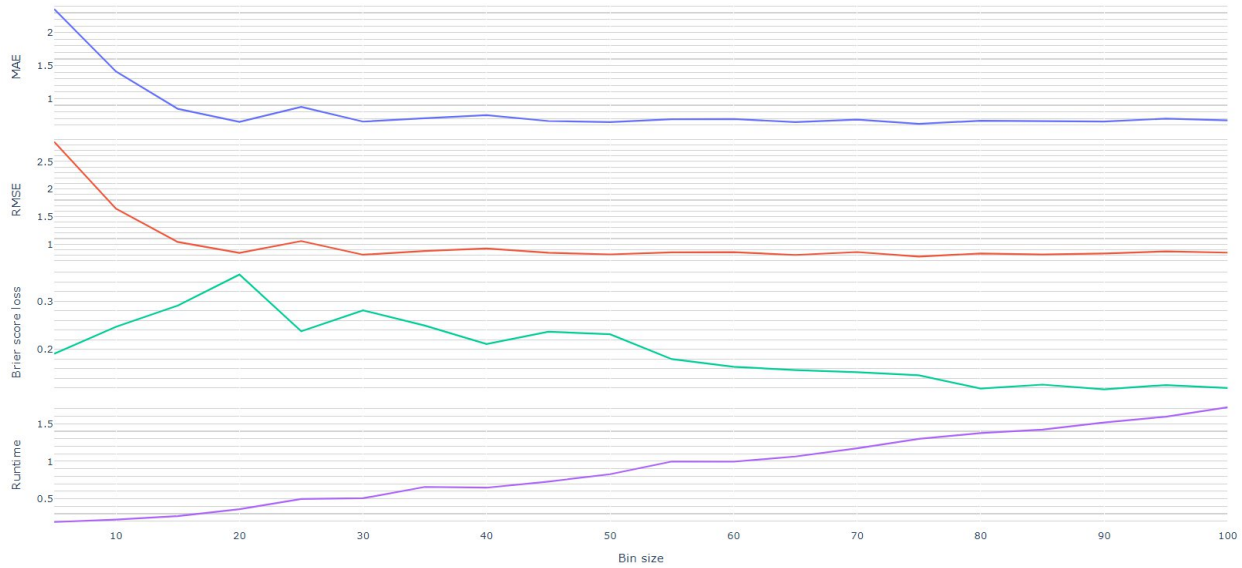

Figure 42: Effect of bin size in the probabilistic model on different KPIs (from top to bottom): MAE, RMSE, Brier score loss, and the runtime (of model processing to generate prognostics)

6.4 Use case results for holistic anomaly detection

A common use of digital twins is anomaly detection in operation, which is usually a precursor to fault detection and diagnostics. Figure 43 shows the main data points involved in humidity control in the thermal zone of the AHU: the cooler valve measurement (which, when open, causes dehumidification of the supply air), the predicted value for the cooler valve, and the extract air humidity level (which when outside a specific comfort range, triggers dehumidification or humidification in the cooler, as required). Specifically in January, there are lengthy periods of unlikely system operation, shown by the yellow areas. This is due to the cooler dehumidification turning on even though the humidity level reported through the extract air was already too low. Upon further inspection of the building energy system, it was evident that this faulty system operation was caused by a bug in the software of the programmable logic controller, which is the device that manages the control logic in the AHU. What is particularly noteworthy here is that the specific rule of dehumidification not turning on when the humidity is low was not encoded in the Bayesian Network models. This rule was implicitly learned as a normal operation. Therefore, even though the cooler valve measurement would not cause an alarm when examined by itself, the ability to evaluate the conditional probability of its state in relation to the other system variables allowed the detection of the faulty behavior. Implementing such a check in real-time would allow for quick detection of faults and save energy.

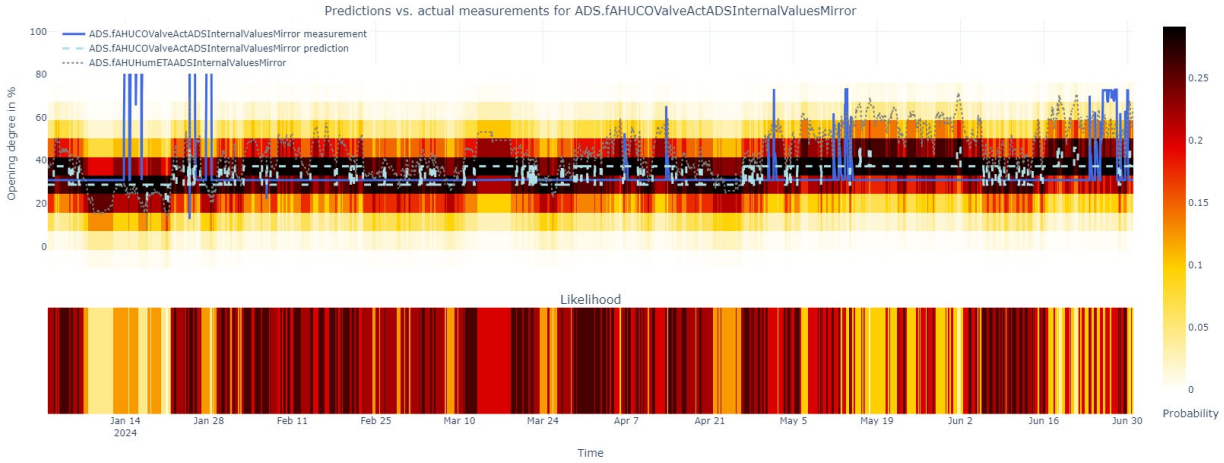

Figure 43: The operation of the AHU cooler is annotated as anomalous, evident by the yellow areas in the heat maps, corresponding to readings with very low probabilities.

7 Technical Workshops (TU Wien, Chalmers)

This year, two 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. The first workshop took place in September, followed by the second in October.

8 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 2024 – Analysis," October 16, 2024. https://www.iea.org/reports/world-energy-outlook-2024

[8] G.F. Cooper and E. Herskovits, 1992. A Bayesian method for the induction of probabilistic networks from data. Machine learning, 9, pp.309-347. https://doi.org/10.1007/BF00994110

[9] 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