Home Energy Management System

ITS8080, semester project

General guidelines for reports

- You may work alone or in pairs.
- Submit the report as one PDF file through the Moodle course webpage or alternatively send it by email to Juri Belikov (juri.belikov@taltech.ee).
- The report can be written in the free format but must contain your name(s) and student code(s) at the top of the first page.
- All the developed codes/links should be sent by email to Margarita Matson (margarita.matson@taltech.ee).

Expected results

1. Textual report [70p]:
   (a) Technical requirements: up to 25 pages.
   Word users: Font size: 12pt; Font family: Times New Roman; Justification: left; Line spacing: multiple to 1,08; A4 page.
   LaTeX users can use the report or article style with default settings.
   (b) PDF file with text (explanations, descriptions, critical analysis, etc.), visualisations, and tables.
   (c) Figures must be clear and have informative attributes (e.g., labels, legends, captions, etc.).
2. Functioning source code [10p]:
   (a) Provide a link to a Git repository with Python Notebook with your code and description.
   (b) Ensure the repository contains an IPython notebook viewer.

How to get the data?

Each student (or a pair) will receive an individual dataset for analysis (three files: `train_test.csv`, `forecast.csv`, and `optimisation.csv`). Data sets will be distributed by Margarita Matson during Practice 4.

Project context

Across the semester, you will progressively develop a full data science workflow for a home energy management system (HEMS), starting from business understanding through data cleaning and visualization to forecasting and simple optimization. HEMS is a smart tool that supervises and controls how energy is produced, stored, and used at home.

With the rise of solar power, electric vehicles (EV), and variable electricity tariffs, households face challenges in balancing energy supply and demand. Without an EMS, much of the solar energy might be wasted, electricity costs may remain high, and battery or EV storage may not be used efficiently. A home EMS ensures smarter use of available resources, supporting both sustainability and cost savings. The primary goals are:

- Reducing annual electricity bills by using energy more efficiently.
- Maximizing self-consumption of renewable solar power to reduce reliance on the grid.

How does a Home EMS work?

Like a traffic controller for electricity. It constantly monitors solar photovoltaic (PV) generation, household demand, battery state of charge, weather forecasts, and electricity prices. Based on this information, the EMS decides whether to use solar power directly, store it in the battery, or draw from the grid. This decision-making is guided by a smart optimization algorithm, tailored to the user's preferences and goals.

Main components

Figure 1 shows the abstract view of a home EMS and its components, which include solar PV panels, a Battery Energy Storage System (BESS), and an intelligent supervising system (software that makes real-time decisions about energy flow). These decisions are made based on defined optimization goals, objectives, and systems operational constraints.

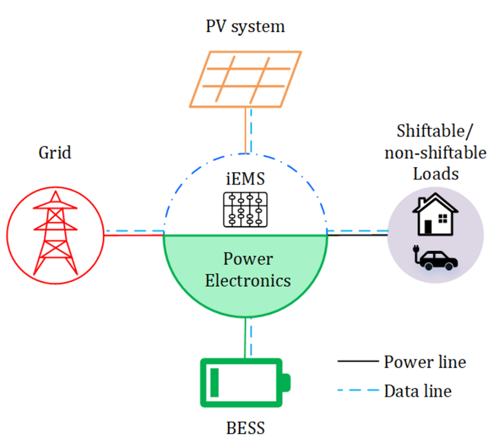

Figure 1: A general overview of the Home EMS.

Tasks overview

1. Introduction to the course & Digital transformation of the energy sector [5pt]

Tips: Inspect and understand provided datasets (variables, units, resolution, etc.). Work with `train_test.csv` file that includes several variables such as `timestamp`, `demand`, `price`, `PV`, and `weather`-related data.

1. Visualize PV generation, demand, and electricity price over a few days.
2. Research how digitalization transforms household energy.
3. Briefly explain in your own words the idea behind working with solar generation data.
   (a) Why is it important?
   (b) How can it be used in private and business sectors?

2. Data science lifecycle (project planning) [4pt]

Tips: Read the project description and understand the main goals.

1. Create a project plan diagram (flowchart) specific to the received dataset, using the data science lifecycle framework.
2. Based on the received dataset, where do you expect the most effort (cleaning, feature engineering, modelling, etc.) in your plan?
3. Identify whether any additional external data sources are required for the analysis or not.

3. Visualisation [5pt]

Tips: Create several figures, ensuring that they follow learned design principles, e.g., incorporating all essential information (labels, legends, etc.) and observing ethical standards.

1. Create timeseries plots and statistical summaries of `demand`, `price`, and `PV` variables.
2. Visualise data using at least two different suitable plots.
3. Which visualisation is the most informative, and why?

4. Data cleaning [10pt]

Tips: Examine carefully which variables are required for your analysis.

1. Work with the `PV_mod1`, `PV_mod2`, and `PV_mod3` variables.
   (a) Identify missing values, outliers, or any inconsistencies in the provided dataset.

(b) Identify the possible types of missing data mechanisms in each case and justify your conclusion.

(c) Use at least three different methods to handle the missing data for the variable `PV_mod1` based on deletion, univariate, and multivariate imputation techniques.

2. Compare quality of each imputed data set using `PV` data.

   (a) Include a numerical summary table (mean, variance, etc.).

3. Visualise before and after for a few representative days.

5. Feature engineering [7pt]

Tips: Work with `demand` and `weather`-related data. You may use simple ranking analysis.

1. Provide a description of the data, including statistics and 2-3 visuals to gain insights.

2. Analyse distributions.

   (a) Are the data normally distributed?

   (b) Are any transformations needed? Apply the latter if relevant.

3. Create at least one new time-related and weather-based feature.

4. Rank features by relevance.

   (a) Explain the rationale behind your ranking. Why do the top-ranked features make sense in the context of the considered application?

   (b) If the engineered feature is poorly ranked, what could be the reason?

6. Time series analysis [5pt]

Tips: Decompose `demand` into trend (T_t), seasonality (S_t), and residuals (R_t).

1. Do the step-by-step additive classical decomposition ($y_t = T_t + S_t + R_t$). Describe each component.

2. During which months/weeks does the seasonal effect appear strongest in your dataset? What could be the reason(s)?

3. Create typical demand profiles.

4. Explain the methodology used to derive these profiles.

7. Statistical modelling [8pt]

Tips: Use only `demand` as the target variable.

1. Stationarise the data, if necessary.
2. Construct, analyse, and describe ACF and PACF plots.
3. Train an autoregressive model ($y_t = \phi(y_{t-1}, y_{t-2}, \ldots, \epsilon_t)$). Use any two ARMA-family models (may be of different orders).
4. Evaluate and compare the performance using normalised RMSE metric.
   (a) Use validation based on the entire training dataset.
   (b) Use the walk-forward validation for the last week in the `train_test.csv` dataset. Split iterations into daily folds.
5. Which model performs better, and why?

8. Machine learning [6pt]

Tips: Use only `demand` as the target variable.

1. Train any suitable advanced ML model (ANN, LSTM, XGBoost, etc.).
2. Provide a table with selected hyperparameters and explain the rationale behind your choice.
3. Compare the developed model to the above statistical model.

9. Forecasting pipeline [8pt]

Tips: Build a reproducible pipeline from cleaning to forecasting. It will be useful for the next task. Work with the `forecast.csv` file.

1. Provide an out-of-sample forecast of hourly demand for 7 days.
   (a) Use a rolling out-of-sample forecasting approach (i.e., forecast day-by-day) with a 24h horizon and 0h lead time.
   (b) Use any suitable forecasting strategy (e.g., direct, recursive, etc.).
   (c) Use developed autoregressive statistical and ML models.
2. Compare forecasts generated by developed statistical and ML models and any two baseline models of your choice (e.g., naïve, drift, ACD, etc.).

10. Models with inputs (features) [7pt]

Tips: Use exogenous inputs (engineered features $f_{i,t}$) to improve ($y_t = \phi(y_{t-1}, f_{1,t-1}, \ldots, \epsilon_t)$) the performance of developed statistical and ML models.

1. Evaluate and quantify performance improvements (use MAE and normalised RMSE metrics) of the model over autoregressive models.

11. Optimal control of storage [5pt]

NB! `Price`, `PV` generation, and `weather`-related data are assumed to be given (ideal forecasts). Use developed models only to forecast the `demand`. Work with the `optimisation.csv` file.

Assume a simple scenario in which homeowners prefer to minimize their annual electricity bill with the help of a home EMS and its smart algorithm. The goal of the optimization is to find the cheapest strategy for the next 24 hours while ensuring the feasibility of the selected solution by satisfying the objective function:

$$\min \sum_{t=1}^{n} Gr_{c,t} - Gr_{P,t}, \tag{1}$$

where $Gr_c \in \mathbb{R}^n$ is the cost of buying electricity from the grid, and $Gr_P \in \mathbb{R}^n$ is the amount of profit the system can earn by selling (injecting) energy to the grid at time t, n is the optimization window size (in this case, n equals to 24 hours), t represents the time step.

Consider a setup with:

- PV system with maximum 5 kW power flow capacity.
- Home battery with 10 kWh capacity, and 5 kW charge and discharge power flow capacity.
- Grid connection with maximum 5 kW power flow capacity.

For the above setup:

1. Forecast demand for the next 24 hours and calculate the optimal control profile.
2. Compare two different profiles for the cases of `PV_low` and `PV_high`.

These steps should help you efficiently carry out your data analysis and exploration task. For each sub-task, write short explanations critically describing main observations and outcomes.

If you have any specific questions or need further assistance, please feel free to ask.

Task #	**Main variables involved**	**train_test.csv**	**forecast.csv**	**optimisation.csv**
1	timestamp, demand, price, PV, weather-related data			
2	Conceptual (all variables considered at project level)			
3	demand, price, PV			
4	PV_mod1, PV_mod2, PV_mod3, PV			
5	demand, weather-related data → engineered features (time-based, weather-based)			
6	demand			
7	demand			
8	demand			
9	demand			
10	demand (target), exogenous inputs: engineered features			
11	demand, price, PV, ESS technical specifications			