AI-Based Modeling for Energy-Efficient Buildings

Solution Report by Ferenc Béres

A1. Personal information

Competition Name: AI-Based Modeling for Energy-Efficient Buildings
Private Leaderboard Score: TBA
Private Leaderboard Place: TBA - in the top 4
Public Leaderboard Score: 0.25719
Public Leaderboard Place: 1st
Name: Ferenc Béres
Location: Budapest, Hungary
Email: ferdzso05@gmail.com
GitHub repository of the solution: https://github.com/ferencberes/kaggle_elias_aifbo

A.2 Background

I am a Research Fellow at the HUN-REN SZTAKI Artificial Intelligence Laboratory (ILAB). I hold a PhD in Informatics, and my primary research areas involve social and cryptocurrency networks. Beyond that, I have more than ten years of working experience with data science and machine learning across both academic and industry settings.

In recent years, I have been involved in some energy industry-related projects. This background provided valuable context for understanding the problem and experimenting effectively throughout the competition. I also relied on the GPU infrastructure available through my research group, which allowed me to train models and iterate at a reasonable pace.

I joined the competition relatively late, starting on October 10th, but remained highly active throughout the final weeks. In total, I submitted 84 predictions to the public leaderboard. I competed individually, without forming or joining a team, and the final solution reflects my own work.

A3. Summary

My solution evolved from initially navigating the large and complex sensor ecosystem to leveraging some insights about the Bosch Budapest Campus, especially the distinction between the working-area building (B201) and the HVAC operating building (B205) as provided in the *metadata.parquet* data file. First, I performed an extensive correlation analysis to identify the most informative sensor categories. Then, I extended the

SimpleAIFBOModel, provided by the organizers, into a **"MultiChannelAIFBOModel"** capable of learning representations from additional sensor channels. The model separately encodes each selected channel before merging these representations with those learned from the chilled-water supply temperature (B205WC000.AM01) and the external temperature (B106WS01.AM54), the only two sensor inputs used in the Bosch example code. Careful channel selection—validated on June and July 2024 data—showed that cooler valve settings (AC21), fancoil room temperatures (B201FC*.AM01), and RC (return circuit?) room temperature sensors (AM02) contributed most to accurate 3-hour-ahead predictions of the target variable (B205WC000.AM02). My final model, the average prediction from my two best-performing submissions, achieved the **best public leaderboard (LB) score of 0.25719**. Below, you can read about my solution in more detail.

A4. Components of the final solution

Baseline Solution

The organizers of the challenge provided example code that demonstrates how to process raw sensor data and generate 3-hour-ahead predictions for the target variable, the chilled-water return temperature (B205WC000.AM02) in their HVAC system. Their pipeline implements the following steps:

- Loading the raw sensor time series and resampling all signals into 10-minute snapshots.
- Generating datetime-related features, including hour of day, day of year, and day of week information, with trigonometric encoding (sin, cos) for the hour and one-hot encoding for the weekday.
- Using only two sensors in addition to date and time: the chilled-water supply temperature (B205WC000.AM01) and the external temperature (B106WS01.AM54). For each sensor, the model receives observations from the last hour (i.e., six resampled snapshots) at the time of prediction.
- A multilayer perceptron (MLP), called SimpleAIFBOModel, that receives sensor and datetime features. The architecture consists of two hidden layers with 128 units each, followed by a single output neuron that predicts the target value 3 hours ahead. They trained this model from January 2025 to April, and kept May for validation.
- The original example code achieved a public leaderboard score of **0.44218**.

However, the provided implementation contained an **error in the calculation of the *day of year* information** (see line 344 in the original `main.py`). After correcting this issue, the performance got even worse (LB: **0.49725**). Removing the *day of year* feature entirely produced a significantly better result (public LB: **0.37440**). I refer to this corrected version of the example code as the **baseline model**, and later I use it in combination with my enhanced model.

Correlation Analysis

My initial goal was to identify a small set of sensors that could meaningfully improve the prediction quality. To do this, I performed a correlation analysis on the 10-minute–resampled data, taking into account the 3-hour prediction horizon. Specifically, I computed Spearman correlations between each sensor and the target variable shifted three hours into the future. To ensure robustness, I calculated the average correlation across three time intervals:

- **March–July 2024:** known spring + summer period, August was excluded as it is not part of the evaluation period (June and July 2025)
- **January–May 2025** (the baseline model's training window), and
- **April–May 2025** (to give more weight to the most recent data).

However, directly adding the highest-correlated sensors to the baseline model did not improve the results. This led me to realize that many sensors, despite high absolute correlations, have large gaps in the data. From that point onward, I restricted my analysis to sensors that met two criteria:

1. **No more than 20% missing values during the test period** (June–July 2025), and
2. **More than two unique values**, to exclude near-constant or binary on/off sensors.

After applying these filters, I recomputed the correlations and ranked sensors by the absolute mean Spearman correlation. The resulting top 5 sensors are listed below:

sensor_id	mean_abs_correlation	std_abs_correlation
B205WC001.AM71	0.681027	0.192017
B205WC000.AM71	0.676862	0.192395
B201FC508_1.VT03_2	0.462754	0.291014
B201FC609_1.VT03_2	0.545173	0.187784
B205WC140.AC21	0.558397	0.130382

Still, when I tried adding these sensors one by one to the baseline model, it overfitted. However, in the top 50 or so most correlating sensors, I saw that there are channels (e.g., VT03_2) that seem to be more relevant than others in terms of absolute correlation.

MultiChannelAIFBOModel – Neural Model for Multiple Sensor Channels

Earlier experiments showed that manually selecting a few sensors from a channel or sensor group did not lead to measurable improvement. On the other hand, the baseline *SimpleAIFBOModel* performed surprisingly well using only two sensors along with datetime features. This motivated me to extend the baseline architecture rather than replace it. I propose the **MultiChannelAIFBOModel**, designed to incorporate additional sensor groups while preserving the strong predictive behavior of the baseline model.

The model consists of the following components:

- **Baseline feature encoder:**
  The model first learns a 128-dimensional hidden representation from the baseline features (date and time + 2 sensors) used in the original *SimpleAIFBOModel*. This ensures that the strong performance of the baseline architecture is retained.

- **Additional channel encoders:**
  The model can optionally receive multiple sensor groups as extra input. For each group, it learns a separate, smaller hidden representation—64 dimensions in my experiments. The reduced size helps limit overfitting and keeps the model close to the baseline predictor.
  In the final solution (see Figure 1), separate 64-dimensional representations were learned for:
  - **39 Cooler valve sensors** (AC21)
  - **410 Fancoil room temperature sensors** (B201FC*.AM01)
  - **23 Return-circuit room temperature sensors** (AM02)

- For channels like AM01, where room identifiers are available (e.g., *201.C.332*) for most of the sensors, I provide **average room-level values** instead of individual sensor measurements. This reduces noise and dimensionality: instead of over 400 individual fancoil readings, the model receives 198 averaged room temperatures, including an extra neuron for sensors without room information.

- **Fusion and prediction layer:**
  All hidden representations—baseline and channel-specific—are concatenated and passed through an additional MLP layer with a 128-dimensional hidden unit. This layer learns to combine the information across channels and produce the final 3-hour-ahead prediction through a single output neuron.

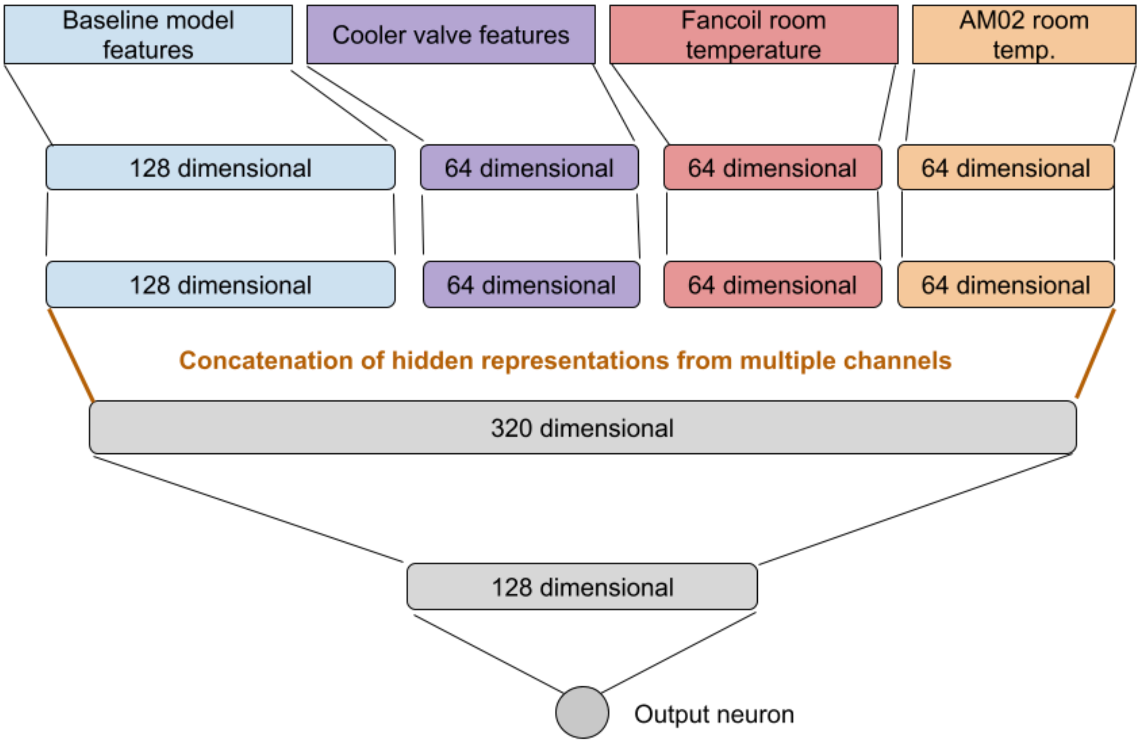


Figure 1: MultiChannelAIFBOModel:. The model first collects information channel-wise, then combines useful descriptors across channels.

Based on my correlation results, first, I added the 39 cooler valve (AC21) sensors as a separate feature group. Unfortunately, the public leaderboard score was still high (0.59570), but when I took the weighted average prediction of this model with 20% weight and the baseline model with 80%, then the combined model (ensemble approach) outperformed previous results **(public LB 0.35430)**.

Final Feature Selection Approach

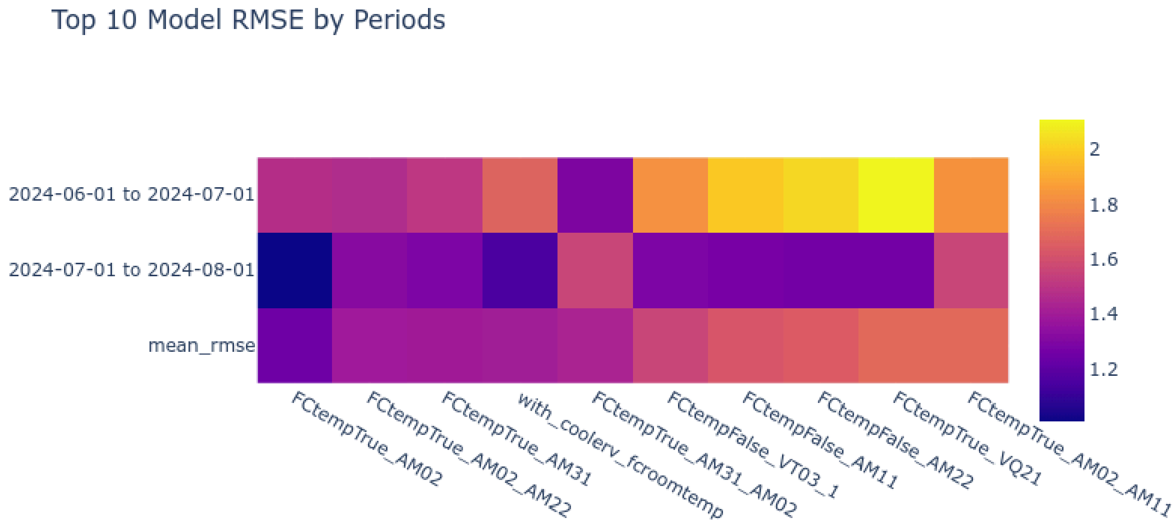
Initial experiments confirmed that extending the *MultiChannelAIFBOModel* with additional sensor channels can meaningfully improve performance. The key question, however, was **which** channels or sensor groups should be added to the model.

My first attempts focused on signals related to human activity inside the building—such as mean CO_2 concentration (AM21), humidity levels (AM45 and AM51), and active setpoint configurations (VT03_2) aggregated by room. However, none of these led to consistent improvements in prediction quality.

To systematically evaluate the usefulness of each potential sensor group, I implemented a testing framework using sensor data from 2024. Following the same protocol as the baseline approach, I trained the *MultiChannelAIFBOModel* on data from **January to May 2024** and evaluated model performance in **June and July 2024**, separately. Sensor groups were deemed *relevant* only if they improved performance in both testing months, ensuring robustness and avoiding overfitting to short-term patterns.

For example, in the figure below, I show the top 10 most promising channel group setups for 2024 June and July based on my measurements. Cooler valves are enabled for every setup.

The setup identifier indicates whether fancoil room temperature sensors were given to the model (e.g., FCtempTrue, with_coolerv_fcroomtemp etc.). At the end of the identifier, you can also see the extra enabled channels (e.g. _AM02, _AM02_AM22 etc.). The results show that the **AM02 channel indeed contains some extra information** compared to previously added channels AC21 and FC room temperature (B201FC*.AM01).



Through this process, I identified three sensor categories that consistently enhanced model accuracy when added to the baseline feature set:

- **Cooler valve sensors (AC21)**
- **Fancoil room temperature sensors (B201FC*.AM01), averaged at the room level**
- **Return-circuit room temperature sensors (AM02)**

Final Submission Details

This section summarizes the composition of my final model and its corresponding public leaderboard (LB) performance. The Table below outlines the improvements achieved as additional sensor groups were incorporated into the *MultiChannelAIFBOModel*. Each new group was added incrementally, meaning previously included groups were always retained.

Group description	Object IDs and description pattern (+)	Sensors	Room avg?	Input dim. to model	LB score in itself	LB score combined (20%-80%)
Cooler valves	*.AC21	39	No	39	0.59570	0.35430
FC room temperature	B201FC*.AM01 + room temp.	410	Yes	198	0.45443	0.31817

More room temperature	*.AM02 + room temp.	23	No	23	0.31145	0.30694
In total:	-	474	-	260		

As discussed earlier, the multichannel model alone did not consistently outperform the baseline when used in isolation—particularly for early-stage configurations such as cooler valves (AC21). However, when I combined predictions from the multichannel model and the baseline model using a **weighted ensemble** (20% multichannel, 80% baseline), the overall performance improved with the addition of each sensor group. The combination weights were tuned through the public leaderboard.

Surprisingly, the final multichain model provided excellent performance in itself on the public leaderboard (0.31145), so in the last few minutes of the competition, I gave more weight to it in the combination. My final submission is the average (50-50%) of the best individual multichain (coolerv+fctemp+AM02) and the baseline model that resulted in **the best 0.25719 public LB score**.

You can find information about the running time on the related GitHub repository.

A6. Interesting Findings

Several key insights contributed to my final placement in the top four. The most impactful was recognizing that adding only a handful of manually selected sensor features brought little improvement, whereas incorporating **entire sensor groups**—with appropriate room-level aggregations—allowed the neural model to learn much richer representations. This shift toward group-based feature learning was essential for unlocking performance gains.

A second important step was establishing a **robust validation framework** using sensor data from 2024. By evaluating sensor groups over the same period of interest (June and July 2024), I could identify which channels were genuinely useful for inferring the chilled-water return temperature (B205WC000.AM02) three hours ahead. It was also important to exclude sensors from the pipeline not steadily available during the test set (June 2025 to July).

Finally, the strategy of **combining multiple predictions** also played a crucial role. Throughout the competition, the simple baseline model (using only two sensors) remained surprisingly strong, while the multichannel model leveraged more than 400 sensors in the end. By averaging these predictions, I achieved a stable and consistently improving ensemble that ultimately produced my top leaderboard score.

Future Research Directions

There are several promising directions that I was unable to explore within the competition timeline, but I believe could enhance model performance.

First, when I was validating extra channel performance for June–July 2024, I found that using **Lasso regression to learn an optimal ensemble** of multiple channel setups yielded excellent results. Training the Lasso weights in June 2024 and evaluating for July 2024 performed remarkably well. However, the learned coefficients did not generalize to June 2025. This suggests that ensemble weights may require **periodic recalibration**, potentially incorporating part of the 2025 data to maintain robust performance.

Next, beyond room and channel information, I did not incorporate additional metadata. When visualizing the average Spearman correlations for sensors in rooms with given room sizes or energy categories, I noticed meaningful patterns. Incorporating these metadata-driven features—or embedding them directly into the deep model architecture—could provide additional structural qualities and improve predictive accuracy.

Most importantly, I'm interested in whether a **knowledge-graph-driven dynamic Graph Neural Network (GNN)** approach is possible for better modeling the HVAC system. The metadata can be used to build a knowledge graph describing relationships between rooms, sensor types, energy categories, equipment hierarchies, and physical layout. Leveraging this structure through a GNN could enable the model to capture spatial and relational dependencies that are difficult to express through tabular features alone. Combining such a model with explainable AI techniques would open up valuable opportunities for interpretability and deeper insights into building operations.

If the organizers are interested in exploring these ideas further, I would be happy to collaborate on any future research or extensions of this challenge.