Development of a Control Concept for Increasing Efficiency and Optimizing Energy Consumption Through Behavioral Pattern Recognition Using Machine Learning in Residential Buildings

Alexander Fritzsche, Armin Lechner (supervisor) and Benjamin Massow (supervisor)

***Abstract*—In the transition to sustainable energy, it is important to adapt consumption to the fluc- tuating availability of power generated by renewable sources. Because a significant portion of the energy for private citizens is used in their own homes this thesis tries to tackle energy usage in residential buildings. Through the use of intelligent control, it is theorized that a reduction in consumption in residential buildings with infrared panel heating is possible without sacrificing comfort and at the same time optimize for the power generated by a photovoltaic system. The idea is to utilize the fast reaction time of infrared heaters to only heat rooms if a person is detected. To achieve this, a system to analyze the usage behavior in a household is devised. It consists of commercially available smart home sensors using the Home Assistant software and accompanying Python API to track data such as temperature, power generation, and if a person is detected. Then an artificial neural network is trained using the fast.ai framework with a combined generated and real dataset. The resulting model shows promise in both predictive control and efficiency gains.**

***Index Terms*—machine learning, fast.ai, energy usage management, home control sys- tems, smart home, Home Assistant, photovoltaic energy management**

A. Fritzsche is a Master student in Mechachatronics & Smart Technologies, MCI, Innsbruck, Austria, e-mail: a.fritzsche@mci4me.at

I. Introduction

To counteract climate change, a shift to sustainable energy management is essential. This transition requires adjusting energy consumption to the fluctuating availability of renewable sources, which now form a growing part of the electricity mix. Photovoltaic (PV) systems are increasingly affordable (a few cents per watt [1]), making them an attractive in-house power source, though their intermittent availability limits practicality. Infrared panel heaters are a cost-effective alternative to heat pumps and gas boilers, though less efficient, converting electricity to heat at a 1:1 ratio, compared to heat pumps' 1:6. To address these challenges, a concept for PV-equipped homes with infrared heating is proposed. Intelligent control can reduce electricity consumption and optimize solar power self-consumption without sacrificing comfort. Infrared heaters' quick response allows rooms to be heated based on detected presence, utilizing smart home technology for control.

Research on residential energy management highlights machine learning (ML) as the most effective method for handling the system's complexity [2]. ML also aids in designing energy-efficient buildings, predicting heating and cooling demands with over 95 % accuracy [3]. Additionally, ML balances grid loads [4] and optimizes electric vehicle charging with

[5] and without [6] PV systems using reinforcement learning. Since 50 % of building energy [7], [8] is used for HVAC, an experiment compared ML-based HVAC control with PID controllers, reducing energy use by 36 % and improving comfort [7]. Another approach uses hot water and room temperatures as energy storage, balancing PV power peaks and achieving slight grid-draw reductions [9]. A third experiment optimized HVAC energy use via Q-learning, reducing energy consumption by 15 % and costs by 17 % [8]. The model adapted well to different comfort zones and seasons. Another study used deep Q-learning for a virtual home with PV and electric vehicles, achieving significant cost savings [10]. Lastly, a big-data approach used IoT sensors to identify energy consumption patterns, applying a J48 decision tree algorithm to suggest optimizations, demonstrating further potential for energy savings [11].

For this paper a Smart Home based on the Home Assistant software [12] is chosen as the basis of implementing a conventional and a machine learning controller. For the latter an supervised learning approach for a random forest and an artificial neural network model is chosen.

II. Methods

As mentioned previously, the Smart Home system uses Home Assistant software running on a Home Assistant Green device, which is powered by an ARM-based Rockchip SoC from Nabu Casa Inc. The system is always updated to the latest stable version. The controller connects to the home network via Ethernet, with a fixed IP address assigned by the router. The interface can be accessed through a mobile app or web browser locally (https://IP:8123) or via Nabu Casa's cloud service.

Device configuration is managed through integrations, which can be installed directly or via the Home Assistant Community Store (HACS). Additional software, called Add-ons, extends functionality. For this project, MQTT Broker and Zigbee2MQTT add-ons are installed to connect Zigbee devices using a USB coordinator.

Sensors and actuators listed in table I manage devices like heating systems, air conditioning, photovoltaic panels, and EV chargers. Millimeter-wave presence sensors detect small movements for accurate occupancy detection which is later used to create patterns for predictions.

TABLE I
List of hardware installed in the smart home network.

type	protocol	amount
controller	-	1
HVAC	Ethernet/IP	1
inverter/powermeter	Ethernet/IP	1
EV charger	Ethernet/IP	3
hot-tub hearter	Ethernet/IP	1
coordinator	Zigbee	1
room thermostat	Zigbee	4
boiler thermostat	Zigbee	1
boiler relais	Zigbee	1
blinds	Zigbee	3
presence sensor	Zigbee	4

The conventional control prioritizes using surplus solar energy. A priority list ensures devices activate sequentially based on available energy and demand. Automation scripts simplify control by breaking tasks into sub-programs, executed every 5 minutes. Energy management balances device activation and deactivation to optimize consumption.

Implemented control automations:

- Climate Control: Adjusts heating/cooling based on season, temperature, and occupancy.
- Infrared Heating: Controlled similarly but with a frost protection function.
- Blinds: Protects against wind and supports heating/cooling by adjusting based on solar
- production.
- Boiler and Hot-Tub: Maintain temperatures and use excess energy for heating.
- EV Chargers: Dynamically regulate charging based on surplus energy, with manual overrides.

The system uses a Lenovo laptop (table II) for training machine learning models, with data logged via a Raspberry Pi and the model deployed on an Nvidia Jetson Nano.

TABLE II
HARDWARE AND SOFTWARE OF THE LAPTOP USED FOR UNSUPERVISED MACHINE LEARNING

component	description
CPU	AMD Ryzen R7 4800H
RAM	2x16 GB SO-DIMM DDR4
GPU	Nvidia GeForce GTX 1650 Ti Mobile
Disk	2 TB SSD
OS	PopOS 22.04 (Linux)

As mentioned earlier, a supervised learning approach is used to train a Random Forest and an Artificial Neural Network, requiring the creation of a dataset. To supplement limited real-world recordings, a synthetic dataset is generated. This section describes the process of creating both datasets. Python 3.12.7 is used for data generation, recording, and model training, with libraries like Pandas and NumPy handling large data processing.

The synthetic dataset simulates realistic daily routines based on occupancy sensor data, reflecting the behavior of a working adult. The generation process begins with real weather and calendar data from January 1. 2000 to August 31. 2023 with historical hourly data sourced from Open-Meteo [13]. Days are initially categorized as *holiday*, *work*, or *off* using the holidays library to get a reference for Austrian holiday dates.

Random vacation days are assigned (e.g., one- or two-week vacations), ensuring vacations are distributed throughout the year. Sick days are generated with a 0.2 % probability and durations between 3 and 13 days. The resulting dataset includes day types (vacation and sick).

Next, room-specific daily routines are created based on day types. For example, on workdays, the routine includes morning activities in the kitchen and bathroom before leaving for work, with evening returns. The output contains five columns representing room types (bathroom, bedroom, kitchen, living_room, study). Presence sensor data is generated minute-by-minute and saved in CSV files.

The datasets are further enriched by integrating hourly weather data for each minute and generating photovoltaic energy production data. Solar energy output is modeled using sunrise/sunset times, cloud cover, and noise factors. Household energy consumption data is also generated with random baseline noise.

To train machine learning models, the datasets are annotated with target temperatures for room heating: a comfort temperature (20 °C to 24 °C), a moderate temperature (19 °C when rooms are vacant), and a low temperature (15 °C for extended absences). Additional annotations include energy usage, heating activity, and time-based splits. The names of the data columns and an example entry are listed in table III.

Real data is recorded using a Raspberry Pi running a Python script that accesses smart home sensors via the Home Assistant API. Data is logged every 5 seconds and saved in daily CSV files. This data is formatted and annotated to match the synthetic dataset, covering the period from November 8. 2023 to March 15. 2024.

For training a Random Forest, the scikit-learn library is used, specifically the RandomForestRegressor. First, the dataset is split into training (80 %) and validation (20 %) sets. The columns are divided into continuous input variables and dependent annotated results. A Random Forest with 100 decision trees and a minimum depth of 5 nodes is initialized and trained over 10 cycles, each with 100,000 random samples, keeping the best result. The lowest validation loss (0.05944) is achieved in the seventh cycle, using Mean Squared Error (MSE) as the loss function.

The model's predictions show a tendency to oscillate between extremes, however, the Random Forest ranks input variables by their influence on the output. Variables *used_power* and below are excluded from the Artificial Neural Network (ANN) training due to their low impact.

For ANN training, the fast.ai framework (built on PyTorch) is used, specifically the Tabular Learner for tabular data. The process has two stages: the first uses the synthetic dataset, and the second retrains the model with real-world data. Both stages split the data into training (80 %) and validation (20 %) sets. A Dataloader prepares the data, with inputs classified as categorical (e.g., room types) or continuous (e.g., temperature). The batch size is set to 8192, limited by a 4GB GPU memory.

The learning rate is determined using a tool provided by fast.ai, resulting in rates of 0.001 for the first stage and 0.0015 for the second. The ANN has two hidden layers (500 and 250 nodes), and outputs are constrained between 15 °C and 25 °C. The Mean Squared Error (MSE) is used as the loss function, and training runs for 40 epochs. The best model is saved as a .pkl file for further training and application.

The ANN runs on an Nvidia Jetson Nano with Ubuntu 20.04, connected to the network via WLAN. A Python script queries sensor data from Home Assistant every minute, processes it, and inputs it into the model to predict the desired room temperature. The prediction is compared with the current temperature to decide if heating should be turned on or off, communicated back via API calls. The process repeats cyclically.

III. Results

To validate the conventional control system, it is put into operation on August 1. 2024, and run under periodic supervision during a test phase until September 30. 2024. At the same time, data is recorded and analyzed using the energy data dashboard integrated into Home Assistant. The months of August and September represent a best-case scenario for commissioning, as high electricity production from the photovoltaic system is expected. Indeed, energy yield is high in these two months. In August, the system delivers an average of 92,2 kWh per day, with a peak value of 123,1 kWh. In September, the average is 65,2 kWh, with a peak of 101,2 kWh, about one-third less. Additionally, in September, there is only one day with energy production over 100 kWh compared to 16 days in August, or roughly half the month. The average consumption over the entire test period remains relatively constant, with an average of 43,9,kWh in August and a slight increase to an average of 46.4,kWh in September.

The self-sufficiency rate for total electricity during the test period is 69,4 %. This is mainly constrained by charging electric vehicles, which, with a peak power of 11 kW, are by far the largest consumers in the smart home. Since they are usually not present on workdays during the periods of highest photovoltaic electricity generation, a significant amount of electricity is exported to the grid and later repurchased in the evening for charging. The boiler and hot tub alone cannot store enough thermal energy to consume 100,kWh by themselves. The air conditioning and infrared heating panels provide a baseline load, but under normal operation, this never exceeds a load of 5,kW, making them unsuitable for buffering large overproduction. However, it is observed that devices like the boiler are almost exclusively active during the day when photovoltaic electricity is available, thus shifting nighttime load to daytime hours.

The self-consumption of photovoltaic electricity during the test period is not particularly high, at 42,4 %, but this is not due to low system efficiency; rather, it is due to the high overproduction in the summer months. Comparing August and September alone, there is already a noticeable increase in the self-consumption rate from 35.4 % to 49.7 %. It is expected that this rate will further increase in the winter months due to the lower solar elevation angle and the corresponding reduction in solar power, as well as the performance loss caused by snow cover. The daily consumption data for the two test months are listed in Tables IV and V.

Additionally the feasibility of training a Machine Learning algorithm to recognize behavioral patterns and thus control room-specific heating is demonstrated. The approach of first generating a synthetic dataset to substitute for missing real data also shows promising results. In this way, it is possible to extend the dataset within the scope of this work, which is limited to three months, and thus deliver meaningful results. However, when comparing the validation loss of the two training processes, it becomes evident that the real dataset is situated at the lower limit in terms of data volume. The loss rate (MSE) increases significantly from 0.067023 to 2.430232 between the two training processes. On the other hand, the room heating control system has an inherently dampening physical effect, which means that a model with more fluctuating output but better predictive properties is preferable to a more stable model.

In Figures 1, 2, 3 and 4 an example for perdiction results compared to annotated training data for each of the rooms can be observed. There the ANN model shows promise in predicting future presence of persons. The gradual ramp up and down of the predicted temperature during a period of high locomotion of people between minute 800 and 1400 is visible in figures 1 through 3. For better results more training and especially more training data is needed which is currently limited by the limited time frame of this paper and the capabilities of the utilized hardware.

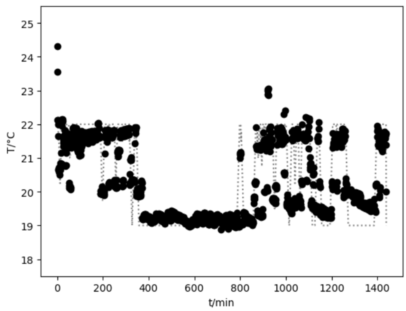

Fig. 1. Comparison of Predictions (datapoints) with annotated data (dashed line) from 28. Dezember 2023 in the eat-in kitchen.

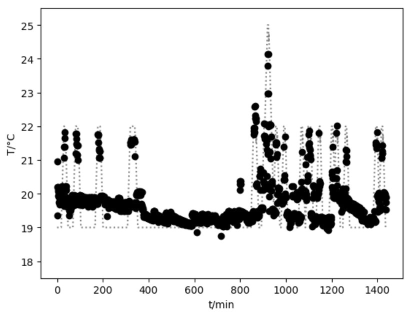

Fig. 2. Comparison of Predictions (datapoints) with annotated data (dashed line) from 28. Dezember 2023 in the bathroom.

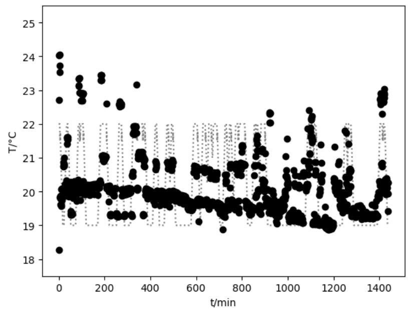

Fig. 3. Comparison of Predictions (datapoints) with annotated data (dashed line) from 28. Dezember 2023 in the bedroom.

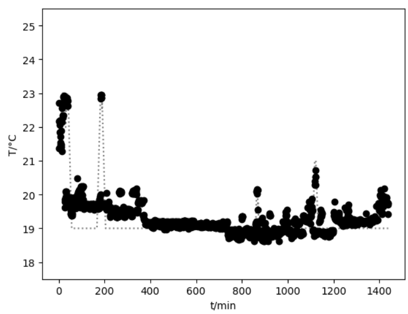

Fig. 4. Comparison of Predictions (datapoints) with annotated data (dashed line) from 28. Dezember 2023 in the study.

IV. Conclusion

In this work, a smart home control system is successfully implemented using the open-source software Home Assistant. Communication with the sensors and controlled consumers is carried out via Ethernet and Zigbee 3.0. A conventional automation system is set up, which controls all connected consumers according to a priority list, adapting their usage to the output of the photovoltaic system. As a result, a solar share of 69,4 % of the total consumption is consistently achieved during a two-month test period. Furthermore, the feasibility of machine-learning-based room heating control is demonstrated. For this purpose, an artificial neural network is trained using the fast.ai framework with both synthetic and real datasets. The goal is to estimate the currently desired room temperature based on behavior patterns determined by presence sensors.

Several approaches exist to further enhance the machine-learning control system. However, this would require more powerful hardware to train on larger datasets. A new solution is also needed for the operational hardware, as the Nvidia DevKit used has reached the end of its product lifecycle and no longer receives official support. Switching to software that does not require CUDA could also be considered, enabling access to a broader range of hardware options.

To generally increase the share of photovoltaics in total electricity consumption, acquiring a battery storage system is deemed beneficial, as it allows energy buffering from day to night or even over several days. This would address the issue where the largest consumers in the system—electric vehicles—are usually not present at the time of peak electricity production.

Finally, the creation of an additional dataset for a new artificial neural network is theorized, which aims to replace the entire conventional control system. Unlike the previous dataset, this one would not be annotated using manual rules. Instead, a genetic algorithm would first determine the optimal combination of active consumers for discrete photovoltaic generation values ranging from 0 kW to 22 kW in 0,5 kW increments. These computed optima could then be used to annotate the new dataset and provide a basis for an improved algorithm.

TABLE III
Columns of the synthetic dataset with an example entry.

column	example value
month	01
day	01
time	18
bathroom	0
bedroom	1
kitchen	0
living_room	0
study	0
weather_temperature	-5.0
weather_cloudcover	100.0
weather_windspeed	4.7
weather_is_day	0.0
solar_power	0.0
used_power	8384.0
last_at_living_room	19.0
last_at_kitchen	101.0
last_at_bathroom	56.0
last_at_bedroom	0.0
last_at_study	205.0
last_at_home	0.0
active_living_room	1.0
heating_living_room	19.0
temperature_living_room	15.1
set_temperature_living_room	23.0
active_kitchen	1.0
heating_kitchen	19.0
temperature_kitchen	15.8
set_temperature_kitchen	21.0
active_bathroom	1.0
heating_bathroom	19.0
temperature_bathroom	9.3
set_temperature_bathroom	24.0
active_bedroom	1.0
heating_bedroom	21.0
temperature_bedroom	18.9
set_temperature_bedroom	21.0
active_study	1.0
heating_study	19.0
temperature_study	17.0
set_temperature_study	20.0

TABLE IV

DAILY ENERGY USAGE STATISTIK FOR THE MONTH OF AUGUST 2024: FROM LEFT TO RIGHT: ENERGY PRODUCED BY PHOTOVOLTAIKS, ENERGY USED, ENERGY EXCHANGED WITH THE GRID, SELF-CONSUMED PERCENTAGE OF PHOTOVOLTAIKS ENERGY AND SELF-SUFFICIENCY PERCENTAGE.

PV / kWh	used / kWh	grid / kWh	self-con. / %	self-suff. / %
81,3	37,2	44,1	35	76
67,8	25,7	42,1	33	86
69,7	49,4	20,3	52	74
92,5	41,7	50,8	33	74
90,3	36,8	53,6	31	76
123,1	24,4	98,6	17	84
81,8	46,5	35,3	45	80
68,6	39,6	28,9	25	43
111,2	86,5	24,6	64	82
119,1	20,2	98,9	15	89
117,7	11,5	106,1	8	82
88,6	7	81,5	5	64
101,8	36,8	65	28	78
102,7	7,3	95,3	5	66
108	15,4	92,6	6	43
99,1	111	-11,9	81	72
95,9	49,1	46,8	40	79
32	27,4	4,6	57	67
43,9	32,4	11,6	48	65
107,2	54,2	53	38	75
111,2	61,8	49,4	26	47
106,2	51,8	54,5	14	29
109,1	47,9	61,1	39	90
101,3	46,9	54,4	36	78
39,6	33,2	6,4	44	52
66,5	45,1	21,4	42	63
114,9	117,5	-2,7	66	65
106,6	28,7	77,9	18	67
106,1	85,4	20,7	72	90
100,6	56,4	44,2	50	89
92,5	27,5	65	24	80

TABLE V

DAILY ENERGY USAGE STATISTIK FOR THE MONTH OF SEPTEMBER 2024: FROM LEFT TO RIGHT: ENERGY PRODUCED BY PHOTOVOLTAIKS, ENERGY USED, ENERGY EXCHANGED WITH THE GRID, SELF-CONSUMED PERCENTAGE OF PHOTOVOLTAIKS ENERGY AND SELF-SUFFICIENCY PERCENTAGE.

PV / kWh	used / kWh	grid / kWh	self-con. / %	self-suff. / %
52,1	65,8	-13,8	89	71
101	40,5	60,5	33	81
95,1	53	42,1	47	84
85,7	31,3	54,4	29	80
54,6	23	31,6	30	72
73,3	30,2	43,1	32	78
101,2	40,1	61,1	35	89
43,9	35,5	8,4	52	65
37,9	31,2	6,7	55	67
64	53	11	42	50
72,8	51,6	21,2	31	43
19,4	43,4	-24	67	30
17	37,7	-20,7	48	22
36,4	27,6	8,8	54	71
86,4	88,2	-1,8	81	80
22,4	23,2	-0,8	59	57
97,3	50,2	47,1	44	86
90,8	27,9	62,9	25	81
72,3	23	49,3	28	88
69,3	91	-21,7	84	64
84,2	60,8	23,4	46	63
78	74,8	3,2	78	81
78,4	41	37,4	46	88
73,3	33,3	39,9	39	86
72,4	41,5	30,9	47	81
34,2	39,8	-5,6	46	40
28,7	31,1	-2,4	57	53
55,3	41,9	13,3	45	60
94,2	97,4	-3,2	77	75
63,9	62,6	1,4	46	47

REFERENCES

[1] H. Ritchie, “Solar panel prices have fallen by around 20% every time global capacity doubled,” 2024.

[2] M. Nutakki and S. Mandava, “Review on optimization techniques and role of artificial intelligence in home energy management systems,” *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105721, 2023.

[3] M. S. Aslam, T. M. Ghazal, A. Fatima, R. A. Said, S. Abbas, M. A. Khan, S. Y. Siddiqui, and M. Ahmad, “Energy-efficiency model for residential buildings using supervised machine learning algorithm,” *Intell. Autom. Soft Comput*, vol. 30, no. 3, pp. 881–888, 2021.

[4] I. Kim, B. Kim, and D. Sidorov, “Machine learning for energy systems optimization,” *Energies*, vol. 15, no. 11, 2022.

[5] N. Kaewdornhan, C. Srithapon, R. Liemthong, and R. Chatthaworn, “Real-time multi-home energy management with ev charging scheduling using multi-agent deep reinforcement learning optimization,” *Energies*, vol. 16, no. 5, 2023.

[6] D. Wu, G. Rabusseau, V. François-lavet, D. Precup, and B. Boulet, “Optimizing home energy management and electric vehicle charging with reinforcement learning,” *Proceedings of the 16th Adaptive Learning Agents*, 2018.

[7] S. Yang, M. P. Wan, W. Chen, B. F. Ng, and S. Dubey, “Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization,” *Applied Energy*, vol. 271, p. 115147, 2020.

[8] O. Kotevska, K. Kurte, J. Munk, T. Johnston, E. Mckee, K. Perumalla, and H. Zandi, “Rl-hems: Reinforcement learning based home energy management system for hvac energy optimization,” *U.S. Department of Energy*, 2 2020.

[9] P. Lissa, C. Deane, M. Schukat, F. Seri, M. Keane, and E. Barrett, “Deep reinforcement learning for home energy management system control,” *Energy and AI*, vol. 3, p. 100043, 2021.

[10] Y. Liu, D. Zhang, and H. B. Gooi, “Optimization strategy based on deep reinforcement learning for home energy management,” *CSEE Journal of Power and Energy Systems*, vol. 6, no. 3, pp. 572–582, 2020.

[11] I. Machorro-Cano, G. Alor-Hernández, M. A. Paredes-Valverde, L. Rodríguez-Mazahua, J. L. Sánchez-Cervantes, and J. O. Olmedo-Aguirre, “Hems-iot: A big data and machine learning-based smart home system for energy saving,” *Energies*, vol. 13, no. 5, 2020.

[12] Open-Home-Foundation, “Home assistant documentation,” accessed: 2024-08-01.

[13] Open-Meteo, “Historical weather data innsbruck,” accessed: 2023-09-07.

Alexander Fritzsche is a Master student in Mechachatronics & Smart Technologies at MCI and is currently working in software developement for production machine solutions with colaborative robots. He also has serveral years of experience in PLC programming and an interest in machine learning.