Proceedings of Machine Learning Research vol 242:1–12, 2024

Adaptive neural network based control approach for building energy control under changing environmental conditions

Lilli Frison LILLI.FRISON@ISE.FRAUNHOFER.DE
Fraunhofer Institute for Solar Energy Systems ISE, Heidenhofstr. 2, 79110 Freiburg, Germany
Systems Control and Optimization Laboratory, Department of Microsystems Engineering (IMTEK), Georges-Koehler-Allee 102, 79110 Freiburg, Germany

Simon Gölzhäuser SIMON.GOELZHAEUSER@ISE.FRAUNHOFER.DE
Fraunhofer Institute for Solar Energy Systems ISE, Heidenhofstr. 2, 79110 Freiburg, Germany

Abstract

Deep neural networks are adept at modeling complex relationships between input and output variables. When trained on diverse datasets, they can understand not just the specifics of individual objects but also the broader principles governing an entire object class. This research applies this principle to building heating control, a domain marked by significant heterogeneity and constant environmental changes, including renovations and changes in user behavior. Our approach involves training the network on a wide range of data instances, enhancing its adaptability to newly distributed data representing unseen scenarios. We find that Transformer-based LSTM architectures are particularly adept for this task as they are able to remember previous tasks' learning. We propose a simple yet effective control algorithm that separates system identification and forecasting from the optimization-based control step. This separation simplifies the control process while ensuring robust performance. In a wide range of simulation experiments, we demonstrate that our "universally trained" neural network control can adjust to changing conditions, thus reducing the need for more complex continual learning techniques. Our results suggest that training neural networks on varied datasets empowers the network with the ability to generalize and adapt beyond specific training instances, which demonstrates their effectiveness in dynamic and heterogeneous environments.

Keywords: Learning based control, adaptive control, building energy control, neural network based control

1. Introduction

The integration and advancement of various heating technologies in buildings present exciting opportunities for reducing carbon emissions and energy consumption. However, these technologies give rise to intricate systems that must operate under diverse and ever-changing conditions, such as weather variations, user behavior, energy availability, and power grid demands. Consequently, effective management of heat generation and distribution systems, along with intelligent control of heating systems under varying circumstances, becomes paramount for optimizing energy efficiency and operational costs. Recent years have seen a growing focus on the smart operation of heating systems within buildings, marking it as a critical research area, involving modern techniques such as model predictive control (MPC), AI and reinforcement learning. Accurate predictions about the future behavior of these dynamic systems are needed to enable well-informed decision-making, helping to mitigate uncertainties and their potential consequences. An essential element is system identification. This process involves two steps: creating precise mathematical models for dynamic systems behavior and calibrating the model parameters. Both steps are cumbersome and require expert's knowledge. Real world processes are under constant change, and model parameters need to be adapted continuously in order to account for changing systems and environmental conditions. Conventional physics-based modeling approaches can be intricate and challenging to calibrate, mainly due to the diversity of building types. Deep Neural Networks (NNs) have emerged as a promising tool for system identification tasks due to their ability to capture complex nonlinear relationships between input and output variables. There is a growing connection between traditional control theory and modern data-driven technologies, leading to new possibilities for efficient and adaptive control systems. However, capturing the intricate interplay of various influencing variables in physical systems often demands the depth of nonlinear systems modeling. The constraints inherent in linear methodologies underscore the necessity to explore advanced techniques like machine learning, with a specific emphasis on neural networks, to effectively navigate and manage the complexity and dynamic nature of nonlinear systems.

1.1. Related works

RNNs are widely regarded as the primary neural network approach for modeling nonlinear dynamic processes due to their inherent sequential data handling, memory retention, and flexibility in capturing complex temporal dependencies. Modern RNN-based process models such as Long Short-Term Memory (LSTM) Hochreiter and Schmidhuber (1997), the simplified version Gated Recurrent Unit (GRU) Cho et al. (2014a), and Encoder-Decoder Sutskever et al. (2014) have established themselves in this regard. Cho et al. (2014b) proposed an Encoder-Decoder architecture, where two RNNs are used sequentially to capture input sequences with varying lengths and dependencies. Zhang et al. (2021) demonstrated that for dynamic processes with many long-term dependencies, Encoder-Decoder-based RNNs outperform LSTM/GRU-based RNNs. In recent years, Transformer architectures, which incorporate various types of attention modules, have shown promising performance in time series prediction. Lim et al. (2021) developed a sophisticated Transformer architecture called the Temporal Fusion Transformer (TFT) for general time series forecasting, surpassing other state-of-the-art methods. Gölzhäuser and Frison (2023) compared the performance of five cutting-edge NN time series forecasting architectures, both state-of-the-art and custom-designed, for predicting building room temperature.

With the aim of overcoming the drawbacks of model-based control while still benefiting from its performance, examples can be found in the literature where NNs were trained with data from an MPC-controlled system, like in the work of Kumar et al. (2018). Drgoňa et al. (2018) proposed a framework for the synthesis of control strategies that mimic the behavior of optimization-based controllers, where also NNs were used. A similar approach was proposed by Frison et al. (2020) comparing different Supervised Learning NN architectures for learning optimal heat pump control using imitation learning from optimized MPC-generated strategies. This Supervised Learning approach was also compared to a control strategy based on Deep Reinforcement Learning. The latter was investigated in more detail by Rohrer et al. (2023). Afram et al. (2017) modeled a residential house with NNs and used them in a supervisory MPC for control. The focus of Wu et al. (2019) article is on the design of MPC systems for nonlinear processes, which predict nonlinear dynamics using an ensemble of RNN models. The goal is to integrate these systems into a control algorithm. In their article, Bonassi et al. (2022) discuss the integration and evaluation of various RNN structures in MPCs, particularly concerning stability guarantees, safety checks, and consistency with the physical system of the RNN models. For an application in building climate control, Li and Tong (2021) used an Encoder-Decoder RNN model to develop an MPC for controlling a building climate control system, demonstrating good convergence and stability. For the same problem, Ellis and Chinde (2020) used Encoder-Decoder models and argued that these can be easily constructed from a model identification perspective.

1.2. Contribution

In this research paper, we illustrates the potential of training a NN on a variety of data covering various system settings in a continual learning setting. Due to their deep architecture, a single NNs is able to model the thermal behavior of various buildings and buildings under changing environmental conditions. For this task, we derive a Attention Encoder Decoder LSTM, which represents a light-weight version of the famous TFT. To integrate the complex NN into an optimization-based control algorithm, we derive a spline-based surrogate algorithm, which decouples the control algorithm and the prediction model.

2. Neural network-based predictive control algorithm

2.1. Control problem formulation

We solve the following output control problem, which regulates the system output to follow a defined reference value while penalizing changes of the control variable. u is the control input, y is the control output and p are disturbances, which could similarly be modelled as uncontrolled inputs. N_p is the past estimation horizon, N is the prediction horizon and $N_u \leq N-1$ is the control horizon after which $u_k = u_{N_u}$ for $N_u < k \leq N-1$. f^{NN} is a blackbox function that given past trajectories of the control output, control input and disturbances, predicts the next output variable as the output of a neural network.

$$\min_{u_0,\ldots,u_{N_u}} \sum_{k=0}^{N} \|y_k - y_{\mathrm{ref}}\|_Q^2 + \sum_{k=0}^{N_u} \|u_k\|_R^2 + \sum_{k=1}^{N_u} \|u_k - u_{k-1}\|_{R^\Delta}^2 \tag{1a}$$

$$\text{s.t. } y_0 = \tilde{y}_0 \tag{1b}$$

$$y_{k+1}, \ldots, y_{k+N} = f^{\mathrm{NN}}\left(y_{k-N_p}, \ldots, y_k, u_{k-N_p}, \ldots, u_{k+N}, p_{k-N_p}, \ldots, p_{k+N}\right) \tag{1c}$$

$$y_k \in \mathscr{Y} = [y_{\min}, y_{\max}] \quad \forall k = 0, \ldots, N \tag{1d}$$

$$u_k \in \mathscr{U} = [u_{\min}, u_{\max}] \quad \forall k = 0, \ldots, N-1 \tag{1e}$$

To design the neural network-based predictive control (NNPC) algorithm, we separate the control aspect from the system identification and forecasting components. By treating the latter as a black box, we enhance the modularity of the system. This separation not only facilitates easier modifications and updates to either component without impacting the other, but also significantly improves the explainability of each part.

2.2. System identification and forecasting module

Our goal is to develop a sophisticated NN architecture for time series forecasting in a system identification during control context, ensuring sufficient generalization ability to react to new, previously unseen inputs arising during the optimization step of the control algorithm and adapt to changing environmental conditions. We use a NN to predict the system behavior for sequence length future N timesteps into the future $y_{k+1}, \ldots, y_{k+N}$. For this, the NN is fed with input data x composed of three parts, which is made of past data and in some cases, future data, cf. Figure 1. The first part consists of all past data over a history horizon of sequence length past N_p timesteps. The second part consists of known forecasted values and future temporal context. And the third part of the input is the control input $u_{k+1}, \ldots, u_{k+N}$, which is assumed to be implemented for the forecast. The Temporal Fusion Transformer (TFT) is a very sophisticated NN architecture, but it comes with the drawback of high computational complexity and resource requirements. To preserve the TFT core strengths while enhancing its memory efficiency and inference speed, we have developed an Attention Encoder-Decoder LSTM (AEDL) architecture. This tailored model is designed to optimize the balance between computational resources and predictive performance, capitalizing on the inherent advantages of TFT while addressing its limitations in memory and processing speed. For the AEDL architecture, we add Input Attention (IA) and Temporal Multi-Head Attention (TA) mechanisms to an Encoder Decoder LSTM. One LSTM is responsible for receiving the past input time steps and the other one for receiving the future input time steps. The IA modules enable the network to attend to a specific part of the input during a certain timestep, respectively to a specific feature. With the TA, the model can additionally attend to certain time points in the input data, as it also is the case for the TFT. Instead of being directly fed into the first LSTM, the past input timesteps $x_{k-N_p}, \ldots, x_k$ are passed through an IA module, one after another, beforehand. Afterwards, the TA calculates the attention to certain timesteps. The future input timesteps $x_{k+1}, \ldots, x_{,k+N}$ are fed into another IA module initially. The IA's output for each future input timestep is concatenated with the TA's output before being passed through the second LSTM. A single-layer FFN computes the final output predictions $y_{k+1}, \ldots, y_{k+N}$. Both IA modules compute a Layer Norm after the Multi-Head Attention calculation. The AEDL architecture is depicted in Figure 1.

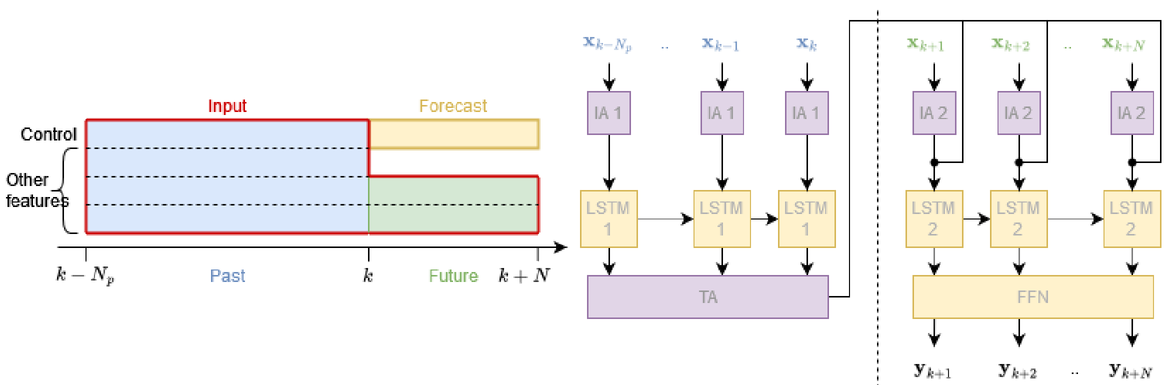

Figure 1: NN input data (left image) composed of known past data (blue), known future data (green) and future control input (yellow). AEDL Architecture (right image).

2.3. Control Strategy

When solving an optimization problem directly with large and complex neural network dynamic functions, potential issues include high computational demands, slow convergence, numerical instability, and difficulties in accurately computing gradients. Additionally, in real operation in a changing environment it may be necessary to include several networks for different purposes such as novelty detection and memory retention. To address these challenges, instead of directly integrating neural network predictions into a derivative-based optimization algorithm—which requires computing derivatives of the neural network model—we employ a surrogate-based optimization approach. This approach (see Algorithm 1) first constructs a piecewise polynomial approximation (PPA) of the objective function, e.g., by using 4th-order splines, treating the neural network prediction as a black box or oracle call. We then optimize the PPA. Building upon the control input u_{k-1} implemented in the preceding timestep, we establish an interval $[u_{k-1} - d, u_{k-1} + d]$ to identify the outputs corresponding to the sampled inputs using the NN prediction. The interval $[u_{k-1} - d, u_{k-1} + d]$ prevents a flattering input behavior and is motivated by objective function of the formulated control problem. Using the sampled data points, we construct a spline-based piecewise polynomial surrogate model of the objective function.

Input: Past horizon N_p, prediction horizon N, control horizon N_u, past control input $u_{k-1}, \dots, u_{k-1}$, past output $y_{k-1}, \dots, y_k$, past and forecasted external influences $p_{k-N_p}, \dots, p_{k+N}$, control interval step d
Output: Optimized control inputs u_k
for *each mpc iteration k* **do**
1. Select a set of M possible control sequences $(u_k^i, \dots, u_{k+N_u}^i)$ within the interval $[u_{k-1} - d, u_{k-1} + d]$
2. For each $i \in M$ query NN model with $u_k^i, \dots, u_{k+N_u}^i$ (and parameters) to get output $y_{k+1}^i, \dots, y_{k+N}^i$ (Eq. 1c)
3. For each $i \in M$ compute objective function value $\hat{y}^i$ (Eq. 1a) using sampled data points
4. Construct PPA using sampled points $(u_k^i, \dots, u_{k+N_u}^i, \hat{y}^i),\ i \in M$
5. Optimize PPA using derivative-based optimization to find optimal control input u_k^*
6. Set $u_k \leftarrow u_k^*$, run the controlled system and advance one step

end

Algorithm 1: NNPC using PPA

To keep the computational burden low, we select a small M such as 10 and set N_u as low as 1 or 2, leading to a low dimensional spline surface model as a surrogate.

3. Application to thermal building control

3.1. Building simulation framework for data generation

The data that we used in this work to train, validate and test the proposed heat pump control strategy is obtained from an open access building thermal energy simulation framework provided by Fraunhofer ISE, called I4B (Intelligence for Buildings) Zhang et al. (2024). The tool can be used for generating a wide range of synthetic thermal building simulation data. Additionally, it serves as a comparison framework for advanced heat pump control strategies. It simulates a simplified building with a single room which is heated by an underfloor heating system. The setup and functionality of I4B is shown in Figure 2. The buildings are defined by the following parameters: Living area, room height, transmission losses, ventilation losses, thermal capacity of the building as well as information about the windowed areas. An air-water heat pump is used to heat the respective building. The heating system is defined in terms of heat pump type and refrigerant mass flow. The simulation framework provide different reference controllers such as traditional MPC or a tunable heating curve control, which serves as a standard reference controller in this field.

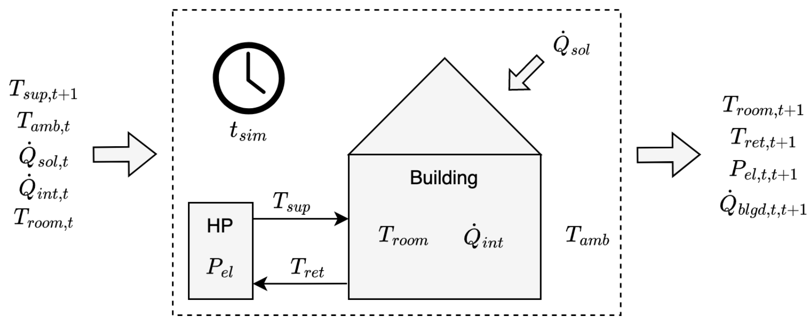

Figure 2: I4B simulation framework.

We obtain the parameters for the buildings from the web tool TABULA [1]. This tool provides information for generically specified buildings in European countries which are categorized by region, period of construction, state of renovation and building type. The renovation states for German houses are defined as follows:

- Renovation state 1 describes the building in its original state without any further renovation. We refer to this as SOC, meaning state of construction.
- Renovation state 2 corresponds to a commonly renovated building, referred to as ENEV (named after the German energy saving regulation 'Energieeinsparverordnung' from 2001).
- Renovation state 3 describes the building after an advanced renovation, referred to as KFW (named after a funding program of the German bank 'Kreditinstitut für Wiederaufbau').

The forecasting model was trained on seven buildings, each from varying construction periods and potentially renovated, totaling approximately 445,000 samples. Three additional buildings, with around 190,000 samples each, were used for validation and for testing. Training data were generated through simulations that incorporated diverse heating curve controllers to reflect the unique thermal characteristics of each building. We obtain ambient temperature and solar irradiance from the online weather data acquisition tool PVGIS[2]. The building simulation framework subsequently calculates the solar gains that impact the building's thermal behavior. Additionally, an internal heat gains profile can be specified, which includes thermal contributions from inhabitants' body heat and heat emitted by electrical devices. These internal and external gains are critical factors in modeling and managing the building's thermal dynamics. For the forecasting model, data from 2011-2013 is used for training and validation, with 2015 data for testing. For the control algorithm, 2015 data is employed for validation and 2017 data for testing.

1. https://webtool.building-typology.eu
2. https://joint-research-centre.ec.europa.eu/pvgis-photovoltaic-geographical-information-en

3.2. Forecasting model hyperparameter optimization

Hyperparameter optimization (HPO) is conducted through the Optuna framework Akiba et al. (2019), employing a Tree-structured Parzen Estimator Bergstra et al. (2015) for sampling and a Hyperband Pruner Ozaki et al. (2020) to terminate unpromising trials. Based on the conducted HPO experiments, we fix the sampling rate of 30 min, the past time horizon of 48h (past sequence length $N_p = 96$) and the future time horizon of 4h (future sequence length $N = 8$) in order to focus the comparison on the type of architecture. Selected hyper parameters for AEDL and TFT are summarized in Table 1.

	AEDL	**TFT**			**AEDL**	**TFT**
Hyperparameter			**Metric**			
Layers l	1	1	$\overline{MAE}_{\text{val}}$	(K)	0.14	0.12
Hidden size h	99	31	$MAE_{\text{max,val}}$	(K)	2.63	2.87
Hidden cont. size h_c	-	26	$\overline{MAE}_{\text{test}}$	(K)	0.55	1.38
FFN size f	-	-	$MAE_{\text{max,test}}$	(K)	2.43	13.55
Attention heads a	4	7	Storage size	(MB)	0.48	0.52
Dropout $drop$	-	0.222	Inference speed	(ms)	4.42	46.28

Table 1: Results of forecasting model architecture comparison on the validation dataset, along with the HPO-found model hyperparameters.

The AEDL model employs LSTM cells without warm-up steps. At each prediction step, the internal states of the LSTMs are reset and reconstructed using the new input data.

3.3. Experimental results

For the TFT, we utilize the implementation provided by the PyTorch Forecasting module. The custom-built AEDL is implemented in PyTorch. Standard SciPy functions are employed for optimization and spline interpolation.

3.3.1. BEST FORECASTING MODEL & CONTROL PERFORMANCE

The control performance is evaluated based on comfort deviation (CD) from the target temperature, electrical energy consumption (E_{el}), and heat pump efficiency, indicated by the coefficient of performance (COP). The COP represents the ratio of supplied thermal energy for heating the house to the electrical energy required by the heat pump. CD is assessed in terms of both positive and negative deviations from the target value, and is further analyzed for greater meaningfulness by considering both average and maximum values. Ideally, both maximum and average negative CD and E_{el} should be low, while COP should be high. In comparing the forecasting performance of AEDL and TFT, the TFT model achieves superior performance on the validation dataset but requires significantly more inference time, cf. Table 1. Additionally, in terms of prediction performance on the test data set during the control experiments, AEDL outperforms TFT. While the mean MAE for the AEDL rises to 0.55 K, the corresponding value for the TFT is almost three times as high, with 1.38 K. This causes the ANN HP control algorithms to produce unrealistic and unusable results when incorporating the TFT as the room temperature forecasting model. For example, the control simulation results for building 3 are shown in Table 2. The results indicate that, although the AEDL

	$CD\uparrow_{max}\ (K)$	$\overline{CD\uparrow}(K)$	$\overline{CD\downarrow}(K)$	$CD\downarrow_{min}(K)$	$E_{el}(kWh)$	COP
NNPC-AEDL	4.17	0.17	-0.12	-1.48	9853.45	3.34
NNPC-TFT	15.17	6.73	-0.07	-2.12	22675.62	2.16

Table 2: Control performance of the neural network-based predictive controller (NNPC) using AEDL vs. TFT as forecasting model with maximum and average negative and positive comfort deviation, electrical energy consumption and heat pump coefficient of performance.

model performs slightly worse on the training data validation set, it generalizes significantly better to the new application data compared to the TFT model. The data used to train the models differ from the data encountered during the application of the adaptive HP control algorithm: The training data originate from the standard rule-based heating curve controlled simulations, whereas the application data are generated on-the-fly by the control algorithm itself. This improved generalization of AEDL may be due to the TFT architecture's higher sensitivity to changes in data characteristics or the need for more training data to fully leverage the TFT architecture's potential.

Table 3 depicts the results for the three different buildings from the validation set compared against a standard heating curve control.

	1949..1957	ENEV	1969..1978	SOC	2010..2015	KFW
Metric	**RB-B**	**NNPC**	**RB-B**	**NNPC**	**RB-B**	**NNPC**
$CD^{\uparrow}_{max}$ (K)	3.03	2.20	3.26	1.98	7.26	6.43
$\overline{CD^{\uparrow}}$ (K)	0.94	0.05	1.10	0.05	1.40	0.17
$\overline{CD^{\downarrow}}$ (K)	-0.01	-0.06	0.00	-0.11	0.00	-0.13
$CD^{\downarrow}_{min}$ (K)	-0.84	-0.80	-0.46	-0.81	-0.67	-1.53
E_{el} (kWh)	3427.76	3056.46	11578.40	9948.30	2003.30	1709.47
COP	4.12	4.33	3.1	3.33	4.60	4.90

Table 3: Comparison between NNPC-AEDL and reference standard heating curve controller RB-B (rule-based basic) on the three buildings from the validation set.

3.3.2. ADAPTABILITY TO CHANGING ENVIRONMENTAL CONDITIONS

Our research motivation was to assess if NN-based control is capable of handling changing operational conditions as well as changing environmental conditions. In this subsection, we show the results of experiments conducted to answer this question by assessing the performance on several different scenarios. In all of them, heat pump control is accomplished with AEDL as forecasting module on one of the validation buildings and a building with construction period 1969-1978 in state SOC is selected. Note that the standard rule-based controller lacks the capability to automatically detect and adapt to changes, making it unsuitable as a reference.

Scenario 1: Increase of internal gains due to change of user behavior. For this experiment, we increase the internal gains $\dot{Q}_{int}$ by multiplying each value of the respective time series with a value uniformly sampled from the interval $[1, 2]$, from a certain point of time t_{change} onward. As a possible real-world scenario, this could be related to a new person moving into the household. Figure 3 shows the transition between the different internal gain profiles at t_{change}. We can observe that NN-based control is able to adapt the control to the new environmental conditions after a short time period.

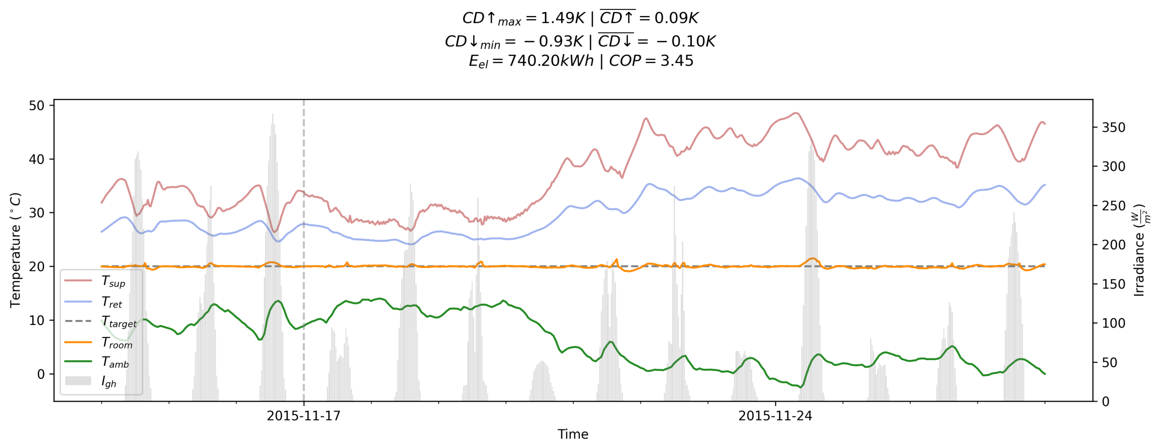

Figure 3: NN-based control simulation with change in internal gains. This figure shows the results over a period of 2 weeks, with higher internal gains from a certain point of time t_{change} onwards. The plot shows the heat pump's supply and return temperature T_{sup} and T_{ret}, the ambient weather conditions in terms of ambient temperature T_{amb} and solar global horizontal irradiance I_{gh}, as well as the progression of the room temperature T_{room}. t_{change} is marked with a vertical dashed line.

Scenario 2: Building renovation. In order to assess the NN-controller's ability to handle big changes made to the controlled environment during runtime, we simulate the renovation of a building from state SOC to state KFW at a certain time point t_{change}. The results are depicted in Figure 4. Directly after t_{change}, we observe high positive comfort deviations $CD\uparrow_{max}= 2.81K$. With advancing time, this behavior recedes and reaches the usual control performance, as soon as the model's past time horizon (48h) does not include t_{change} anymore.

Scenario 3: Varying target room temperatures. Besides changes in the controlled environment, we also consider adaptions of the target room temperature T_{target}, which influence the thermal behavior of the building. Figure 5 shows the control simulation results with a setback from $T_{target} = 20\,°C$ to $T_{target} = 17\,°C$ for a duration of one week. In the real world, this behavior could e. g. be related to a time during which the inhabitants are on vacation but do not want the building to cool down completely. We can observe that the NN control manages the transitions between the different target room temperatures with a delay of up to approximately $12\ h$. The transition from lower to higher T_{target} is delayed more and also results in a subsequent room temperature overshoot ($CD\uparrow_{max}= 2.67K$).

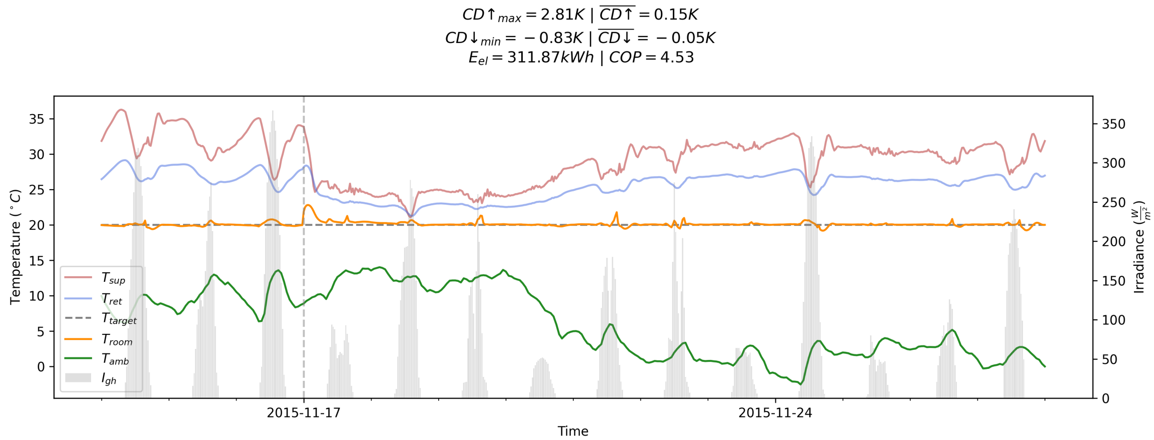

Figure 4: NN-based control simulation with building renovation scenario.

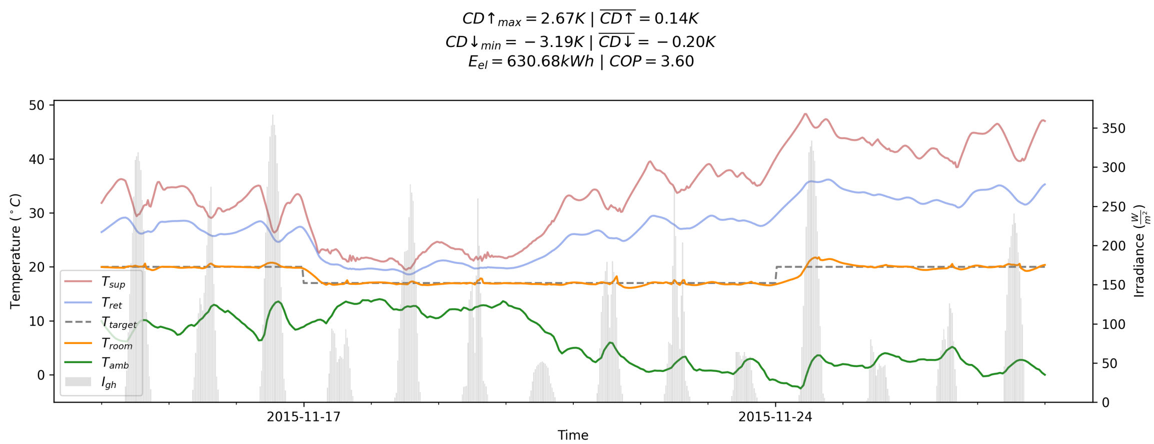

Figure 5: NN-based control with target room temperature change.

4. Conclusion

This paper introduces an methodology for optimizing building heating control in a dynamic environment by leveraging sophisticated deep neural networks trained on varied datasets. Employing a novel combination of Transformer-based LSTM architectures, it enhances adaptability to changing conditions. Additionally, the introduction of a spline-based surrogate-control algorithm distinguishes system identification from the control process itself. Future advancements may include integrating the forecasting module directly into the control algorithm and integrating a continual learning module into the control pipeline. Such an enhancement would enable the neural network controller to continuously learn and adapt, effectively absorbing new information from a wide array of real-time situations ensuring that the learning process remains dynamic and responsive.

Acknowledgments

The authors acknowledge the financial support by the German Federal Ministry of Education and Research in the project 'AI4HP' (project code: 01IS21081A).

References

Abdul Afram, Farrokh Janabi-Sharifi, Alan S Fung, and Kaamran Raahemifar. Artificial neural network based model predictive control and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings*, 141:96–113, 2017.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A hyperparameter optimization framework. `https://optuna.readthedocs.io/en/stable/`, 2019.

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. *Proceedings of the 12th Python in science conference*, pages 13–20, 2015.

Francesco Bonassi, Marco Farina, Jianhao Xie, and Riccardo Scattolini. On recurrent neural networks for learning-based control: Recent results and ideas for future developments. *Journal of Process Control*, 114:92–104, 2022.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *arXiv preprint arXiv:1409.1259*, 2014a.

Kyunghyun Cho, Bart Van Merriënboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *arXiv preprint arXiv:1406.1078*, 2014b.

Ján Drgoňa, Damien Picard, Michal Kvasnica, and Lieve Helsen. Approximate model predictive building control via machine learning. *Applied Energy*, 218:199–216, 2018.

Matthew J. Ellis and Venkata Chinde. An encoder-decoder lstm-based empc framework applied to a building hvac system. *Chemical Engineering Research and Design*, 160:508–520, 2020.

Lilli Frison, Sweetin Paul, Torsten Koller, David Fischer, Gianluca Frison, Joschka Boedecker, and Peter Engelmann. Hardware-in-the-loop test of learning-based controllers for grid-supportive building heating operation. *IFAC-PapersOnLine*, 53(2):17107–17112, 2020.

Simon Gölzhäuser and Lilli Frison. Comparison of different deep neural networks for system identification of thermal building behavior. *Journal of Physics Conference Series*, 2600(7):072008, 2023. doi: 10.1088/1742-6596/2600/7/072008.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Steven Spielberg Pon Kumar, Aditya Tulsyan, Bhushan Gopaluni, and Philip Loewen. A deep learning architecture for predictive control. *IFAC-PapersOnLine*, 51(18):512–517, 2018.

Yinghao Li and Shuo Tong. Deep reinforcement learning for hvac control: A review and future directions. *Applied Energy*, 286:116466, 2021.

Bryan Lim, Serim Onur Arik, Nicolas Loeff, and Thomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4): 1748–1764, 2021.

Yoshihiko Ozaki, Yuki Tanigaki, Shuhei Watanabe, and Masaki Onishi. Multiobjective tree-structured parzen estimator for computationally expensive optimization problems. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, GECCO '20, page 533–541, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371285. doi: 10.1145/3377930.3389817. URL https://doi.org/10.1145/3377930.3389817.

Tobias Rohrer, Lilli Frison, Lukas Kaupenjohann, Katrin Scharf, and Elke Hergenröther. Deep reinforcement learning for heat pump control. In *Science and Information Conference*, pages 459–471, 2023.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems, 27*, 2014.

Zhongqiang Wu, Aniruddh Tran, Diego Rincon, and Panagiotis D. Christofides. Machine learning-based predictive control of nonlinear processes. part i: Theory. *AIChE Journal*, 65, 2019. doi: 10.1002/aic.16729.

Baohe Zhang, Lilli Frison, Thomas Brox, and Joschka Bödecker. Constrained reinforcement learning for safe heat pump control, from simulation framework to algorithm design, 2024. Submitted to CDC 2024.

Jun Zhang, Yue Zou, Di He, and Guanghui Sun. A review on the recent developments of sequence-to-sequence models. *Artificial Intelligence Review*, 54(4):2357–2395, 2021.