14th IEA Heat Pump Conference
15-18 May 2023, Chicago, Illinois

Frost Detection with Neural Networks: Determining Necessary Sensors to Predict Optimal Defrost Initiation Time for Air Source Heat Pumps

Jonas Klingebiel[a,*], Paul Salomon [a], Christian Vering [a] and Dirk Müller [a]

[a]Institute for Energy Efficient Buildings and Indoor Climate, Aachen, Germany

Abstract

Air Source Heat Pumps (ASHPs) are the most common heat pump type in Europe's residential buildings. To increase the energy efficiency of ASHPs, a main research field focuses on defrosting management. Currently, researchers showed that optimal defrosting initiation time (ODT) exists, which exhibits great potential to improve operational efficiency. However, ODT depends on multiple factors such as ASHP operation (e.g., compressor RPM) and ambient conditions (e.g., relative humidity). While mapping all correlations between ODT and all relevant factors can be accomplished with artificial neural networks (ANN), gaining sufficient test-bench data is time-consuming. When combining ANNs with reinforcement learning (RL) the data can be automatically generated on-site. A key aspect for the successful realization of RL is the determination of necessary sensors to detect frost under dynamic ASHP operation and varying ambient conditions. This work studies the applicability of different sensor sets to predict frost. Therefore, we use a heat pump model with valid frosting and defrosting behavior. The model is calibrated with test bench data. The results indicate that commonly available sensors in heat pumps are suitable for robust frost detection. Using only the ambient and evaporation temperature, the RL agent can separate frosting behavior from heat pump control and improves energy efficiency by up to 9.4 % compared to conventional time-controlled defrosting.

Selection and/or peer-review under the responsibility of the organizers of the 14th IEA Heat Pump Conference 2023.

Keywords: defrost initiation; self-optimizing control; artificial neural network; reinforcement learning; simulation

1. Introduction

The federal government of Germany has set itself ambitious and legally binding climate protection goals: Greenhouse gas emissions must be reduced by 65% until 2030 and by 88% until 2040, relative to 1990 levels. [1]. Since heat pumps are an resource-efficient option to convert electricity from renewable energy sources to heat, the Federal Environmental Agency attributes a key role to heat pumps in achieving the goals [2]. Currently, Air-Source Heat Pumps (ASHPs) make up 82% of heat pumps installed in Germany's building sector (2021), and their market share rises continuously [3] as they are easier to install and retrofit. Thereby ASHP are more cost-effective while performing only marginally worse than Ground Source Heat Pumps (GSHPs) [4].

During the heating operation of ASHPs, frost may form on the evaporator's fins due to the temperature difference between the heat-transferring surface and the ambient air, depending on the ambient conditions. The frost deteriorates the evaporator capacity and, thus, the heat pump efficiency. To maintain safe and efficient operation, defrosting operations are performed. Wang et al. experimentally show that optimal defrosting initiation time (ODT) exists [5]. Further, they conclude that ODT varies significantly depending on ambient conditions and ASHP operation. However, simple defrosting strategies such as time-based defrosting (TBD) are widely used in commercial applications [6] because of their ease of implementation. In TBD, a defrosting

* Corresponding author.
E-mail address: jonas.klingebiel@eonerc.rwth-aachen.de

operation is performed at a fixed time interval, ranging typically between 60 to 90 minutes [7]. TBD inevitably leads to sub-optimal defrost initiations, so-called "mal-defrost phenomena" [8], which comprise either too early or too late defrost initiations.

To limit the inherent mal-defrost losses of TBD, several attempts have been made to develop demand-based defrosting (DBD) strategies. DBD strategies aim to enhance the defrost initiation by determining correlations between measurable operational parameters and the current state of frosting on the ASHP. Recent works study the application of artificial neural networks (ANNs) [9], [10]. ANNs seem promising for the complex control task due to their capability of capturing complex system relations [11]. Wang et al. [9] propose a supervised learning approach using convolutional neural networks (CNNs). The researcher use samples from a time-based defrosting approach as labeled data for the supervised learning task. The data was manually filtered for mal-defrost operations leading to a data set without apparent mal-defrost phenomena. The trained CNN reached a root mean squared error (RMSE) of 7.2 % and avoided mal-defrosting for most of the investigated test cases. However, the authors constituted that the quality of the labeled data limits the quality of the CNN defrosting strategy. Generating labeled data in sufficient quantities is often a core limitation in ANN applications.

To support this research, we considered the application of Reinforcement Learning (RL) algorithms for DBD in a previous simulation study [12]. In difference from pure ANN approaches, RL allows for learning independent of labeled data since it autonomously discovers the inherent patterns in a dataset. Thus, an RL agent does not impose the limitation of high-quality data but gradually evolves by learning from previous experiences. The investigated Deep-Q-Network (DQN) Agent achieved an average improvement of 5.6% over TBD for a 24h dynamic interval and avoided mal-defrosting operation. However, we used the air-side pressure drop as a frost indicator, among other sensor data, which might not be available in real-world application. In this context, this parameter represents a comparatively direct correlation to frost mass but is difficult to determine experimentally.

Therefore, this paper studies the applicability of different sensor sets to predict frost formation. The paper is organized as follows: In the 2. section we present the basics of RL, and the algorithms used to analyze the results. In the 3. section we describe the case study system, our implementation approaches, and the experiment design. In the 4. section we present the results, which are discussed in the 5. section in more detail. In the 6. section we give a concluding summary as well as inspirations for future work.

2. Methodology

2.1. Reinforcement Learning

RL is a model-free, self-optimizing control algorithm in which an agent interacts with an environment without providing it with instructions on how to act correctly. However, the agent receives feedback on the quality of its actions in the form of a reward r. The goal of the agent is to maximize its immediate and future rewards. Therefore, actions leading to higher rewards are reinforced during the learning process. Based on experience, the agent derives a policy π which maps an action a to each environment state s. The state s is a set of features/sensors representing the environment and includes all information upon which the agent chooses the action. With increasing training, the agent's policy converges toward the optimal policy. Figure 1 illustrates the agents' interactions with the environment.

In Q-Learning, a standard reinforcement learning algorithm, the values of state-action pairs (s, a) with respect to the reward signal are established. In basic Q-Learning the relation between (s, a) and value is stored in a table. While this works well for simple systems with limited states and actions [13], complex problems make use of function approximators. Deep Q-Networks (DQN), a state-of-the-art RL algorithm, use artificial neural networks (ANN) as function approximators [14].

Many design principles have been published over the years, enabling stable ANN training in dynamic environments and improving training data efficiency. As part of our last study, we used two principles: a target network and a replay buffer. The target network is a second ANN, which is used to calculate the Q-values. Its trainable parameters are frozen for a fixed number of interactions. Unlike the main Q-network, the target network's parameters are not trained but periodically synchronized with them. Using a target network prevents instabilities (or even divergence) during training that may arise from rapidly changing policies. The second principle is the use of a replay buffer for experience replay. This overcomes two issues when combining RL with ANNs. First, the data set of RL is non-stationary since the RL algorithm constantly learns new behaviors, but ANNs need stationary data sets. Second, classical ANNs see training samples independent from each other but the training data of RL is sampled from a sequence of correlated states. The replay buffer serves as a growing but stationary data set, breaking the correlation between data samples during training. When RL is

applied to slow-responding thermal systems, including past observations is important because an important design principle (the Markov property) requires that future states depend only on the current state [20].

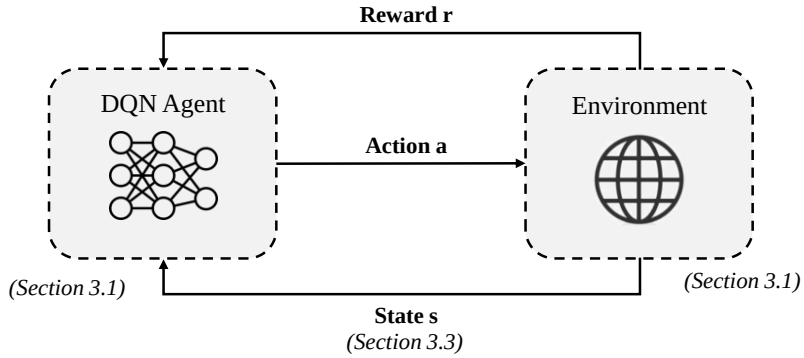

Fig. 1. Schematic illustration of reinforcement learning.

2.2. Feature Importance

Machine learning models can be interpreted by assessing feature importance. To quantify a feature's importance, the model's prediction error is calculated after it has been permuted. As a result of this procedure, the relationship between the feature and the target is broken, thus causing the model score to drop, indicating the model's dependence on it. The feature is considered "important" if randomizing its values increases model error since the model relies on the feature to make predictions. A feature is "unimportant" if shuffling its values leaves the model error unchanged. [15], [16]

3. Experimental design

In this section, we describe the case study system, our implementations in Python and Modelica, and the different configurations of our experiment.

3.1. The Case Study

In the conducted case study, a DQN agent is implemented as a defrost controller and applied to a dynamic heat pump simulation model. The simulation model and the RL agent are described below.

3.1.1. Simulation Model

The dynamic simulation model of the ASHP is implemented in Modelica [17] using Dymola [18] and TIL Library [19], [20]. A detailed description of the calibrated simulation model can be derived from [12]. In the following, the essential principles of the model are outlined.

At the core of the ASHP model is a finite-volume model of the evaporator. The control volume is discretized into several small volumes (cells), and the conservation equations for energy, mass, and impulse are solved for each cell. To model the effect of frost on the heating capacity, the two dominant loss mechanisms are incorporated into the evaporator model:

- **Thermal Resistance**: Frost acts as a thermal insulation layer between the air and the fin. The thermal resistance is a function of frost density and frost layer thickness.
- **Hydraulic Resistance**: Frost reduces the air-side cross-sectional area. For a constant fan power, the air volume flow decreases with increasing frost layer thickness.

To determine the described mechanisms, the frost density and the frost thickness must be modelled sufficiently accurately. For this purpose, the convective water mass flow from the moist air to the frost is divided into frost densification and frost thickening by applying literature correlations. The evaporator model is embedded in a model of the refrigerant cycle with lower complexity to preserve high simulation speeds. The complete refrigerant cycle model is calibrated using measurement data from an ASHP test bench. To enable the dynamic operation of the heat pump model a compressor control is implemented, which adjusts compressor

power to match a set value for supply temperature. Further, a superheating controller regulates a constant superheat at the evaporator outlet.

The resulting heat pump model takes the ambient temperature, the relative humidity, the condenser capacity, and the supply temperature of the heat sink as model inputs as illustrated in Figure 2. Dynamic profiles are generated with AixLib library [21] to simulate real-world scenarios with varying ambient conditions and heat demands. This study uses a building model of a 120 m² single dwelling in Berlin as a case study.

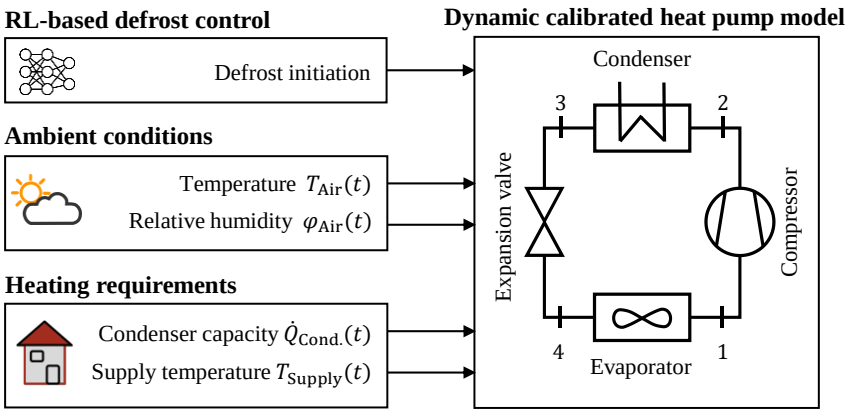

Fig. 2. Inputs of dynamic heat pump simulation model.

3.1.2. Reinforcement Learning Agent

In the following, the design of the control problem is formulated. We use a DQN Agent with the extensions outlined in section 2. The agent interacts with the ASHP model every 100 seconds (simulation time). Every interaction step consists of taking an action, observing the environment, and receiving a reward. The agent decides whether defrosting should be initiated (**action** a). Thereby the agent's action space is discrete:

$$a = \begin{cases} 1\text{: initate defrost operation} \\ 0\text{: stay in heating operation} \end{cases} \tag{1}$$

While defrost initiation is controlled by the RL agent, defrost termination is not part of the agent's control domain. The defrosting operation is terminated by a threshold value with regard to the frost mass.
The agents' state s, which is composed of several features, is varied in this study. The selection of suitable state spaces is described in section 3.3. The *COP* of the system can serve as a **reward** function, but its value is strongly dependent on ambient conditions. Therefore, selecting COP as reward would penalize the agent for thermodynamic demanding ambient conditions. To eliminate this deficiency, we use the Carnot efficiency as a reward function:

$$r = \frac{COP}{COP_{\mathrm{Carnot}}} \tag{2}$$

The Carnot efficiency normalizes the *COP* to the physical optimum. As a result, ambient conditions do not directly affect the reward signal. The described interaction approach exhibits temporarily delayed rewards: The agent has to learn that occasional defrosting operations positively impact the future reward trajectory even though it receives a penalty in the short term.

3.2. Implementation

Figure 3 shows the overall framework in which the agent and the environment (heat pump simulation model) interact. The simulation model written in Modelica is exported as Functional Mock-up Unit (FMU).

FMI is an open-source standard for the simulation of dynamic models generated in Dymola or Simulink in other frameworks such as Python. The FMU is embedded in Python with the Python library FmPY [22]. In order to develop reinforcement learning agents, the FMU is embedded into an OpenAI Gym environment [23]. These provide standardized interfaces in the field of RL research to test and compare algorithms efficiently. The algorithms used are provided by the Pytorch-based stable-baselines library [24]. The library provides tested state-of-the-art RL algorithms, a comparatively user-friendly API, and many recurringly needed evaluation and backup methods.

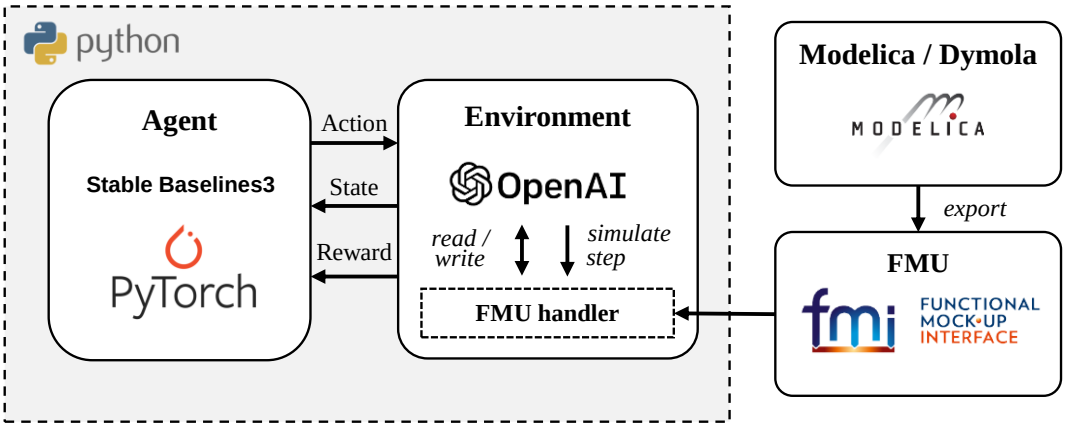

Fig. 3. The software modules used in this study and their functional relationships.

3.3. State space variations

In this study, we investigate the influence of different state spaces on the performance of an RL-based defrosting controller. In many cases, the environment is not fully observable, preventing the agent from developing an accurate picture of its environment. The features must contain sufficient information about frost accumulation and its impact on the operation. To develop a reasonable strategy, the agent must separate between phenomena caused by frost formation and other unrelated influences. For this purpose, sensors which directly measure the existence of frost while having few external influences are advantageous. However, these sensors are usually not installed in conventional heat pumps and may be expensive and unreliable.

For example, frost mass or air-side static pressure difference over evaporator are good frost indicators, but they represent an additional asset in terms of cost and reliability. On the contrary, the evaporating temperature is robust and inexpensive to measure but contains only an indirect link to frost mass. Other parameters, such as fan or compressor speed, influence the evaporation temperature, complicating the control problem. In general, a small state with high-quality information leads to faster convergence. Considering the abovementioned limitations, we selected the following features:

- Ambient temperature T_{Air}
- Prior action a_{t-1}
- Evaporating temperature $\mathrm{T_{Evaporator}}$
- Evaporator outlet temperature $\mathrm{T_{Evaporator,out}}$
- Condensing temperature $\mathrm{T_{Condenser}}$
- Air-side static pressure difference over evaporator Δp_{Air}
- Electrical power consumption of the compressor $\mathrm{P_{el,compressor}}$

Section 4.1 describes the influence of frost and external disturbances on each parameter. The features were grouped into state spaces based on expert knowledge. Tab. 1 displays five state spaces that were investigated in this study. The state spaces are increasingly difficult (from #1 to #5) since number and information quality are reduced successively. While the first state space (#1) contains variables that exhibit a direct link to frost mass (air-side pressure difference Δp_{Air}), the last one (#5) only includes the ambient temperature and the prior action, which significantly complicates the control problem. All state spaces contain the air temperature T_{Air} and the last performed action a_{t-1} since they are easy to determine and have proven to be relevant parameters for the agent's decision. Additionally, except for #5, each state space contains the evaporating temperature.

Since the air-side pressure difference Δp_{Air} has to be determined by additional sensors, it is only utilized for state space #1. From #3 on, the compressor power $\mathrm{P_{el,compressor}}$ is also discarded since measuring this parameter is costly. As a result, only standard sensory is used from #3 - #5.

Table 1. Investigated state spaces

#1	#2	#3	#4	#5
T_{Air}	T_{Air}	T_{Air}	T_{Air}	T_{Air}
$a_{\mathrm{t-1}}$	$a_{\mathrm{t-1}}$	$a_{\mathrm{t-1}}$	$a_{\mathrm{t-1}}$	$a_{\mathrm{t-1}}$
$T_{\mathrm{Evaporator}}$	$T_{\mathrm{Evaporator}}$	$T_{\mathrm{Evaporator}}$	$T_{\mathrm{Evaporator}}$	
$P_{\mathrm{el,compressor}}$	$T_{\mathrm{Evaporator,out}}$	$T_{\mathrm{Evaporator,out}}$		
Δp_{Air}	$P_{\mathrm{el,compressor}}$	$\mathrm{T_{Condenser}}$		
	$\mathrm{T_{Condenser}}$			

Besides the sensor values listed above, historical sensor values are also included in the state space. In addition to the current sensor value at time t, the last 39 historical sensor values are provided as input to the agent.

3.4. Training

The training is divided into two phases. The goals of the training stages are explained in the following:

- **Pre-training**: Pre-training aims to train the agent to cope with constant ambient conditions. After pre-training, the agent should recognize patterns between ambient conditions and frost growth rate. Therefore, the agent should be capable of adjusting the time point of defrost initiation concerning the ambient conditions in a static case. The agent is required to generalize as the ambient conditions are random and nonrepetitive. Pre-Training is terminated when a certain reward threshold is reached.
- **Main training**: Main training aims to train the agent in a dynamic environment. After the main training, the agent exhibits a reasonable defrosting strategy for dynamic ambient conditions and heat loads. The agent is also required to identify conditions in which no frost formation occurs and not to initiate defrosting when these conditions are present. The agent is trained for 150.000 steps, corresponding to 170 days of simulation time.

4. Results

In this section, we present the results obtained with our experiment design.

4.1. System dynamics

In the following, the state spaces' features and their influence on frost are analyzed. Figure 4 displays selected features for a frosting-defrosting cycle. Two scenarios were simulated for constant ambient conditions and heat demands to highlight the influence of frost growth velocity on the corresponding features. The first represents severe frosting conditions, and the second mild frosting conditions. Table 2 summarizes the simulation set values for both scenarios. The frost growth rate is high in the severe frosting scenario due to high absolute humidity. In contrast, the frost growth rate is lower in the "mild frosting" scenario.

The frost growth is reflected in the accumulated frost mass. The gradient is higher for severe frosting than for mild frosting. This relationship can be extracted directly from the signal for the air-side pressure drop: The pressure drop increases faster in the severe frosting scenario. Faster growth of frost causes the air-side cross-sectional area to decrease rapidly so that the pressure drop increases and the volume flow decreases. In the severe frosting scenario, the air-side pressure drop at t = 2000 converges to a limit at 100 Pa. Here, the air-side cross-sectional area is completely blocked. The increase in frost mass is only due to frost densification.

Table 2. Boundary conditions for simulation (see figure 4)

Scenario	Air temperature T_{Air} in °C	Relative humidity φ_{Air} in %	Condenser capacity $\dot{Q}_{\mathrm{cond.}}$ in kW	Supply temperature T_{Supply} in °C
Severe frosting	4	80	7000	35
Mild frosting	-6	80	7000	35

The compressor's electrical power shows similar characteristics: As the frost mass increases, the transferred heat at the evaporator decreases. To maintain a constant condenser capacity, compressor power is increased by the heat pump controller to compensate for the reduction in evaporator capacity. If heat demand, ambient temperature, and supply temperature are constant during the heating phase (frosting), the electrical power is a valid frosting indicator. Under frost-free conditions, the air-side pressure drop is identical for both scenarios and is thus independent of external influences (see figure 4, t=0). In contrast, the compressor's electrical power depends on the heat demand and the supply temperature and thus differs under frost-free conditions.

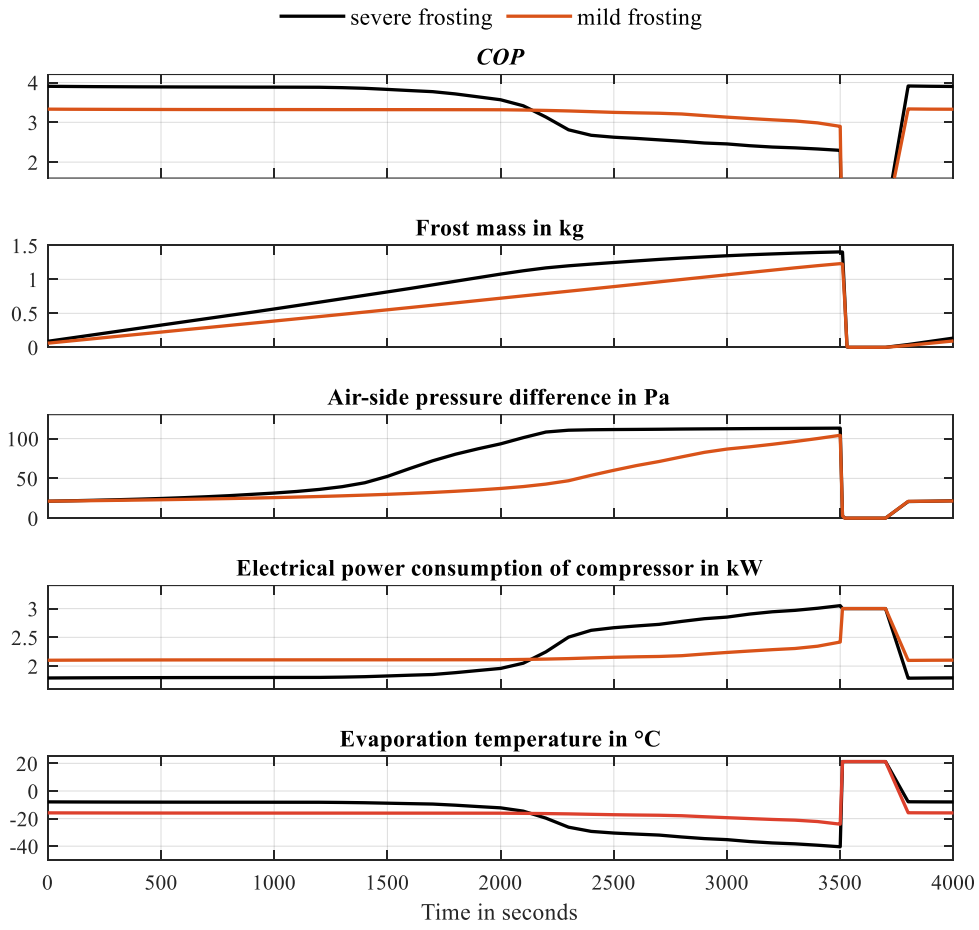

Fig.4. Influence of frost on heat pump system.

Figure 4 also displays the evaporating temperature. The evaporating temperature decreases with increasing frost mass due to the expansion valve control, which regulates a constant superheat. The evaporating temperature shows a qualitatively similar curve to the compressor's electrical power consumption. While the evaporating temperature decreases significantly in the severe frosting scenario, only a slight decrease can be seen in the "mild frosting" scenario. The evaporating temperature differs under frost-free conditions due to the varying ambient temperatures.

In addition to the quantities shown in figure 4, additional features were selected for frost detection. The temperature at the output of the evaporator differs from the evaporation temperature by a constant value (superheat). The condensation temperature correlates with the ambient temperature according to the heating curve and thus has an indirect influence on frost growth.

4.2. State space variations

The agents with the state spaces defined in Table 1 were trained according to the description in section 3.4. After training, the performance of the agents was evaluated for ten randomly selected test periods with varying environmental conditions and heat demands with a length of 72 h. The average reward per step for each test period was calculated to compare the agent's performance. The reward corresponds to the Carnot efficiency (see equation 2). The value was then averaged over the ten episodes.

Additionally, a time-based defrost controller that initiates a defrosting operation every 60 minutes was evaluated. The 60 minutes correspond to a typical value from the literature [7]. Figure 5 displays the average reward per step over the ten test periods for RL and TBD. Additionally, minimum and maximum deviation is displayed.

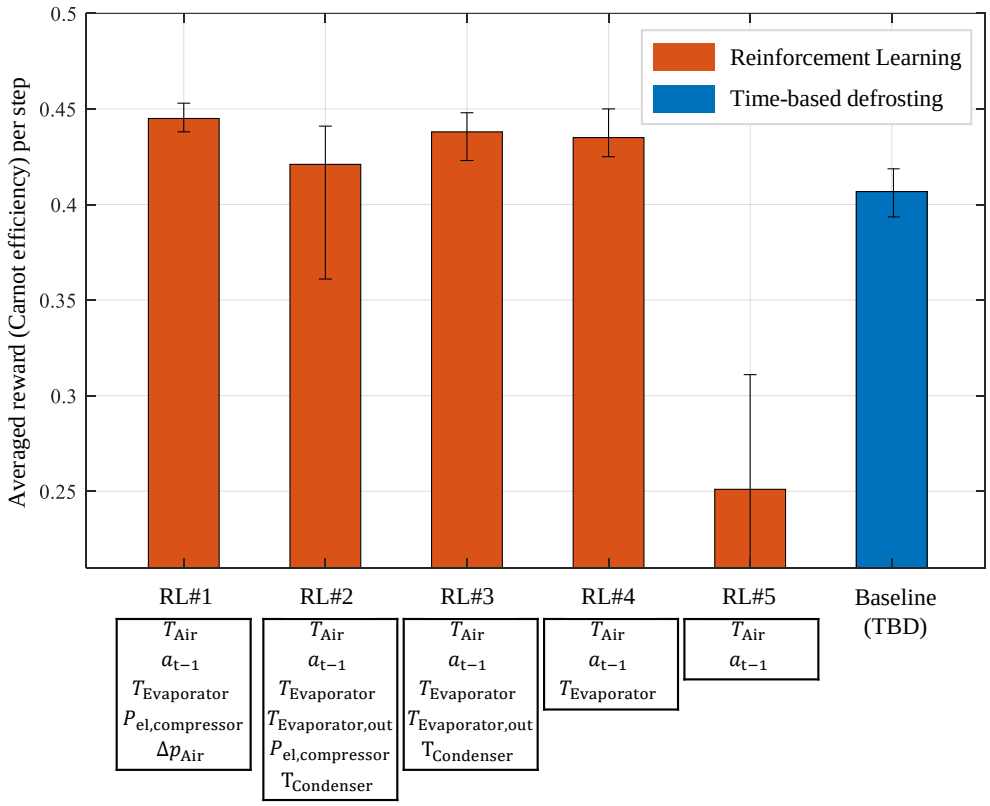

Fig.5. Results of state space variation.

RL with state space #1 (RL#1) achieves the highest reward per step. However, RL#3 and RL#4 achieve comparable scores. While the performance of RL#4 is marginally worse compared to RL#3, this may be due to the randomness of the selected test conditions. The similar performance between RL#3 and RL#4 implies that the evaporator output temperature and condensing temperature do not add beneficial information. Based on the results, we conclude that demand-based defrosting can be accomplished with just the air temperature, the evaporation temperature, and the last action.

RL#2 performs significantly worse compared to RL#3. At first, this seems counterintuitive since RL#2 contains all features of RL#3. The difference between RL#2 and RL#3 is in the electrical compressor power signal, which was classified as a reasonable frost indicator in the previous section. The weaker performance can be attributed to the higher number of features. The additional information provided by the sensor is not valuable enough for the agent to achieve higher performance. Conversely, the additional feature increases the

complexity of frost detection. As a result, the RL#2 agent requires more training time to achieve similar results as RL#3.

RL#5 shows significantly worse performance compared to all other RL agents. The included features do not exhibit sufficient information to perform demand-based defrosting operations. The agent has to develop a defrosting control based on the ambient temperature. However, since frost growth is, among others, dependent on relative humidity and condenser power, the agent cannot establish a reasonable defrosting strategy.

Compared to TBD, the RL agents #1 - #4 achieved significant efficiency improvements. The increase in efficiency between TBD and RL#1 is 9.4 %. The efficiency improvement between TBD and RL#4 is 8.9 %.

4.3. Feature importance

We performed permutation tests in order to measure feature importance. Therefore, each feature was randomly permuted, while the remaining features were kept untouched. After a permutation was performed, the loss of the model prediction was calculated. This was repeated 100 times (for each feature), and the average prediction loss was evaluated.

Figure 6 displays the model loss due to feature permutation for different state spaces. Across the three state spaces presented, there is no dominant feature. While the absolute value of the model error has a low information value, the relative differences within a state space can be interpreted.

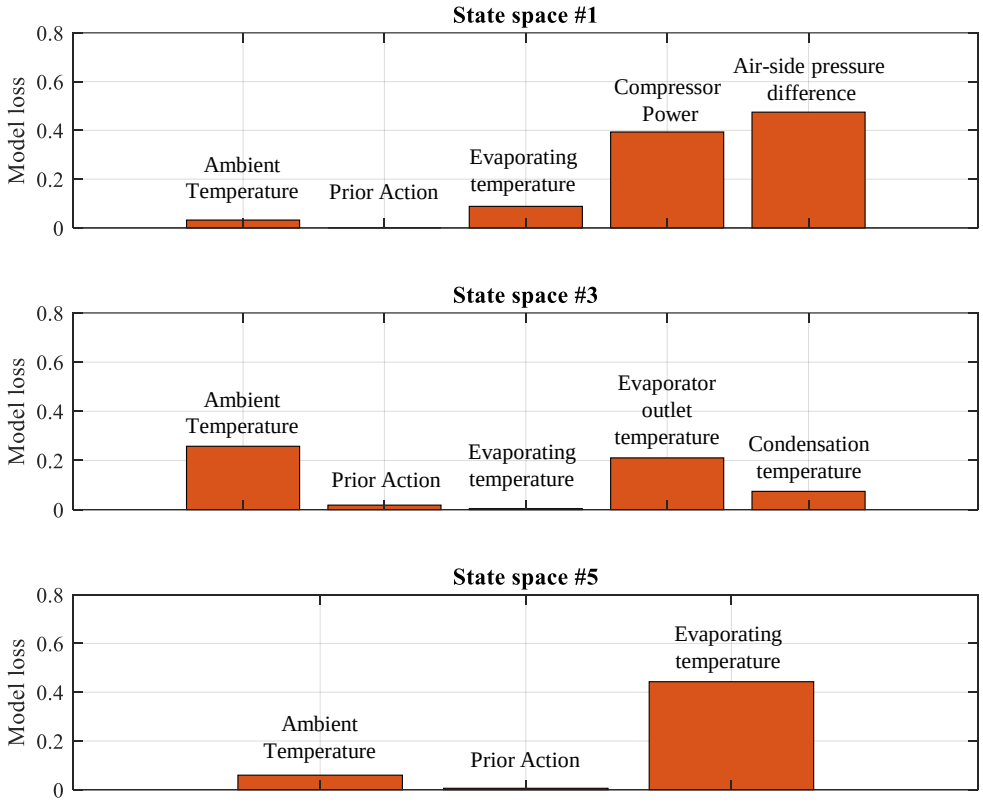

Fig.6. Feature importance of different state spaces.

In state space #1 the air-side pressure difference and the compressor power show the most significant influence on the agent's actions. The remaining features have a minor impact on the agent's decisions. Consequently, the agent has discovered that the two dominant features provide sufficient information to learn a reasonable defrosting strategy.

Ambient temperature and evaporator outlet temperature have the most decisive influence on the agent in state space #3. The feature importance of the evaporating temperature is negligible. The agent has detected that the difference between both features consists of a constant offset (superheating). Thus, the features share identical information as long as the superheat is kept constant by expansion valve control. In state space #5 the

evaporation temperature is the dominant feature since evaporator outlet temperature is not part of the state space. The decisive influence of the ambient temperature is significantly smaller.

5. Discussion

The results show that RL can serve as a demand-controlled defrost algorithm using only standard temperature sensors of the refrigerant cycle as frost indicators. The state space variation and the feature importance evaluation demonstrate that the agent successfully recognizes the sensor values with the highest information quality and assigns the greatest influence to these values in the decision process. Furthermore, we observed that a larger state space leads to a lower performance for the same training time. For this reason, the number of features should be kept reasonably small.

In the calibrated simulation model, the evaporation temperature shows a characteristic drop due to frost. The absolute value of the evaporation temperature is dependent on the ambient temperature. Thus, the agent needs both features to predict defrosting necessity. An improvement could be achieved by applying feature engineering. Here, multiple features are combined into a single value to decrease state space dimension and accelerate convergence speed. With regard to the results of section 4.2, it seems promising to pass the temperature difference between ambient and evaporation temperature to the agent as a single value.

Although the results of this simulation study are promising, the simulation did not take into account several effects that arise in heat pumps in the field and might complicate the control problem significantly. In this study, the fan speed was assumed constant. Since fan speed affects the evaporating temperature, its set value should be implemented as a feature when applying RL to real heat pumps. Furthermore, we did not investigate the effect of controller-related oscillations on the agents' performance. In real-world applications, the frosting influences the controlled system of the expansion valve because system dynamics differ from no-frost conditions. Altered system dynamics can result in an oscillation of the superheat and, thus, of the evaporating temperature. Smoothing the features with respect to time may resolve the issue.

Furthermore, we defined the reward using Carnot efficiency. While this is an excellent quantity to optimize, the computation of the value is subject to variances due to measurement uncertainties. In addition, many frost-unrelated factors influence the Carnot efficiency, e.g., isentropic compressor efficiency. Contrary to the assumption implied in this study, the isentropic efficiency is not constant. Consequently, the reward function will fluctuate in absolute value over the operating range, which is a disturbance factor for the agent. Hence, alternative reward functions that directly relate to frost growth should be investigated.

6. Conclusion and future work

In this paper, we apply a State-of-the-Art RL algorithm to perform demand-controlled defrosting for a calibrated dynamic ASHP simulation model. The results indicate that RL can serve as a demand-controlled defrost algorithm while using only standard sensors of the refrigerant cycle as frost indicators. The agent extracts all necessary information and outperforms conventional time-controlled defrosting by 9.2% by only using two temperature sensors in the state space (ambient temperature and evaporation temperature). We conclude that the selected state space significantly impacts the agent's convergence speed and final energy efficiency. Further, we apply a feature importance analysis to quantify the impact of individual features on the agent's decision.

However, some aspects must be addressed to exploit the full potential for real-word applications. Future work should focus on accelerating convergence speed (e.g., feature engineering) and implementing more on-site effects (e.g., measurement uncertainty) into the simulation model. Additionally, research that investigates the reusability of already trained algorithms to heat pumps with different evaporator geometries could accelerate the deployment of RL algorithms to commercial heat pumps.

Acknowledgements

We gratefully acknowledge the financial support by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the AiF (German Federation of Industrial Research Associations eV) based on a decision taken by the German Bundestag (IGF no. 20701 N / 2).

References

[1] W. Frenz, “Bundes-Klimaschutzgesetz (KSG),” in *Klimaschutzrecht: EU-Klimagesetz, KSG Bund und NRW, BEHG, Steuerrecht, Querschnittsthemen*, W. Frenz, Ed. Berlin: Erich Schmidt Verlag GmbH & Co. KG, 2022, pp. 525–841. doi: 10.37307/b.978-3-503-20687-2.04.

[2] D. V. Bürger *et al.*, “Klimaneutraler Gebäudebestand 2050 - Energieeffizienzpotenziale und die Auswirkungen des Klimawandels auf den Gebäudebestand,” p. 289.

[3] Bundesverband Wärmepumpe e.V., “Absatzzahlen für Heizungswärmepumpen in Deutschland 2013-2019,” *Bundesverband Wärmepumpe e.V. Zahlen & Daten*, Aug. 18, 2020. https://www.waermepumpe.de/typo3temp/yag/11/58/Diagramm_AbsatzzahlenHWP_2013-2019_115809_5e42bdc64.jpg (accessed Aug. 18, 2020).

[4] P. Christodoulides, L. Aresti, and G. Florides, "Air-conditioning of a typical house in moderate climates with Ground Source Heat Pumps and cost comparison with Air Source Heat Pumps," *Applied Thermal Engineering*, vol. 158, p. 113772, Jul. 2019, doi: 10.1016/j.applthermaleng.2019.113772.

[5] W. Wang, S. Zhang, Z. Li, Y. Sun, S. Deng, and X. Wu, "Determination of the optimal defrosting initiating time point for an ASHP unit based on the minimum loss coefficient in the nominal output heating energy," *Energy*, vol. 191, p. 116505, Jan. 2020, doi: 10.1016/j.energy.2019.116505.

[6] M. Song, S. Deng, C. Dang, N. Mao, and Z. Wang, "Review on improvement for air source heat pump units during frosting and defrosting," *Applied Energy*, vol. 211, pp. 1150–1170, Feb. 2018, doi: 10.1016/j.apenergy.2017.12.022.

[7] M. Song, G. Gong, N. Mao, S. Deng, and Z. Wang, "Experimental investigation on an air source heat pump unit with a three-circuit outdoor coil for its reverse cycle defrosting termination temperature," *Applied Energy*, vol. 204, pp. 1388–1398, Oct. 2017, doi: 10.1016/j.apenergy.2017.01.068.

[8] W. Wang, J. Xiao, Q. C. Guo, W. P. Lu, and Y. C. Feng, "Field test investigation of the characteristics for the air source heat pump under two typical mal-defrost phenomena," *Applied Energy*, vol. 88, no. 12, Art. no. 12, Dec. 2011, doi: 10.1016/j.apenergy.2011.05.047.

[9] W. Wang, Q. Zhou, G. Tian, Y. Wang, Z. Zhao, and F. Cao, "A novel defrosting initiation strategy based on convolutional neural network for air-source heat pump," *International Journal of Refrigeration*, vol. 128, pp. 95–103, Aug. 2021, doi: 10.1016/j.ijrefrig.2021.04.001.

[10] Y. H. Eom, Y. Chung, M. Park, S. B. Hong, and M. S. Kim, "Deep learning-based prediction method on performance change of air source heat pump system under frosting conditions," *Energy*, vol. 228, p. 120542, Aug. 2021, doi: 10.1016/j.energy.2021.120542.

[11] S. Kollias, Ed., *Artificial neural networks -- ICANN 2006: 16th international conference, Athens, Greece, September 10-14, 2006, proceedings, part I*, 1st ed. Berlin: Springer, 2006.

[12] J. Klingebiel, S. Göbel, V. Venzik, and D. Müller, "Evaluation of machine learning methods for optimizing the defrosting process of air-to-water heat pumps," *15th IIR-Gustav Lorentzen Conference on Natural Refrigerants*, 2022, doi: 10.18462/iir.gl2022.117.

[13] E. Even-Dar and Y. Mansour, "Learning Rates for Q-Learning," in *Computational Learning Theory*, vol. 2111, D. Helmbold and B. Williamson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 589–604. doi: 10.1007/3-540-44581-1_39.

[14] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," Dec. 2013, doi: 10.48550/arXiv.1312.5602.

[15] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[16] A. Fisher, C. Rudin, and F. Dominici, "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously," p. 81.

[17] Modelica Association, *Modelica\circledR - A Unified Object-Oriented Language for Physical Systems Modeling: Version 3.2 Revision 2*. 2013.

[18] Dymola, *Dassault Systems. Dymola - multi-engineering modelling and simulation. Dymola 2017 (64-bit)*. 2016. [Online]. Available: http://www.3ds.com/products/catia/portfolio/dymola

[19] C. Richter, “Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems,” Universitätsbibliothek Braunschweig, 2008. doi: 10.24355/DBBS.084-200806100200-3.

[20] M. Gräber, K. Kosowski, C. Richter, and W. Tegethoff, "Modelling of heat pumps with an object-oriented model library for thermodynamic systems," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 16, no. 3, pp. 195–209, Oct. 2010, doi: 10/bvtz37.

[21] EBC, *AixLib: A Modelica model library for building performance simulations*. 2018. [Online]. Available: https://github.com/RWTH-EBC/AixLib

[22] GitHub, *FMPy*. 2020. [Online]. Available: https://github.com/CATIA-Systems/FMPy

[23] OpenAI, "Gym: A toolkit for developing and comparing reinforcement learning algorithms." https://gym.openai.com (accessed Aug. 10, 2021).

[24] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," p. 8.