

Article

A Condition-Monitoring Methodology Using Deep Learning-Based Surrogate Models and Parameter Identification Applied to Heat Pumps

Pieter Rousseau *  and Ryno LaubscherDepartment of Mechanical and Mechatronic Engineering, Stellenbosch University,
Stellenbosch 7600, South Africa; rlaubscher@sun.ac.za

* Correspondence: prousseau@sun.ac.za

Abstract: Online condition-monitoring techniques that are used to reveal incipient faults before breakdowns occur are typically data-driven or model-based. We propose the use of a fundamental physics-based thermofluid model of a heat pump cycle combined with deep learning-based surrogate models and parameter identification in order to simultaneously detect, locate, and quantify degradation occurring in the different components. The methodology is demonstrated with the aid of synthetically generated data, which include the effect of measurement uncertainty. A “forward” neural network surrogate model is trained and then combined with parameter identification which minimizes the residuals between the surrogate model results and the measured plant data. For the forward approach using four measured performance parameters with 100 or more measured data points, very good prediction accuracy is achieved, even with as much as 20% noise imposed on the measured data. Very good accuracy is also achieved with as few as 10 measured data points with noise up to 5%. However, prediction accuracy is reduced with less data points and more measurement uncertainty. A “backward” neural network surrogate model can also be applied directly without parameter identification and is therefore much faster. However, it is more challenging to train and produce less accurate predictions. The forward approach is fast enough so that the calculation time does not impede its application in practice, and it can still be applied if some of the measured performance parameters are no longer available, due to sensor failure for instance, albeit with reduced accuracy.

Keywords: condition-monitoring; heat pumps; neural networks; parameter identification



Citation: Rousseau, P.; Laubscher, R. A Condition-Monitoring Methodology Using Deep Learning-Based Surrogate Models and Parameter Identification Applied to Heat Pumps. *Math. Comput. Appl.* **2024**, *29*, 52. <https://doi.org/10.3390/mca29040052>

Academic Editor: Sebastian Skatulla

Received: 11 June 2024

Revised: 1 July 2024

Accepted: 2 July 2024

Published: 5 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Being able to anticipate breakdowns in industrial equipment in advance can reduce the negative impacts caused by unexpected stoppages, especially where complicated logistics are involved for repairs and maintenance. For this purpose, online condition-based monitoring can be applied to reveal incipient faults before breakdowns occur [1]. These techniques can broadly be divided into model-based or data-driven approaches [1–4]. Model-based approaches typically identify faults by comparing outputs predicted by a mathematical model of the system with actual measurements. Inconsistencies between the model outputs and actual measurements are called residuals, which are considered fault indicators [5]. Another approach is to monitor the rate of change in specific key parameters within the physics-based model via parameter identification to detect potential faults [6]. Data-driven approaches typically detect faults via applying signal processing or statistical analysis to the measured data [2]. The ideal procedure includes steps for fault detection, classification, location, and quantification [1,5,7], although many methods focus on detecting faults without identifying the underlying causes. Some purely data-driven fault detection and diagnosis models applied to refrigeration cycles and heating, ventilating, and air conditioning (HVAC) systems include those proposed by Gao et al. [8], Zhang et al. [9], and Du et al. [10].

Model-based methods require a suitable model of the physical system. Models can range from lumped parameter regression models [11] to more detailed physics-based models [12]. Kocyigit et al. [13] developed a fault diagnosis method for vapor compression refrigeration systems based on the pressure-enthalpy diagram. This includes an analysis of the refrigeration cycle to determine the values of pressure and enthalpy in between the different components, including the compressor, condenser, expansion device, and evaporator. Essentially, faults are identified via overlaying the faulty p-h diagram on top of the healthy p-h diagram and looking at the ratios of different variables. Lee et al. [14] used a building energy simulation tool to model the HVAC system of an office building for normal operation and five types of system faults. However, the focus was on the air handling units, and therefore the refrigeration cycle chillers were simulated as lumped models. It could therefore not identify faults within the refrigeration cycle. They used simulated operational data to train a deep neural network classifier in order to identify the individual faults. They did not look at the ability to discriminate between multiple simultaneous faults.

Miyata et al. [3,7,15] developed a model based fault detection and diagnosis system for a HVAC system comprising of four water chillers and water thermal storage tanks. They modelled the water distribution systems with pumps and tanks in some detail, but the refrigeration cycle chillers were modelled via a lumped system curve that provides the Coefficient of Performance (COP) as a function of the load and the condenser water outlet temperature. Faults in the chillers were described as a decrease in efficiency. The models were based on a quasi steady state assumption, although the controllers accounted for the change in variables over time. They used the simulation model to generate a fault database with fault labels, which was then converted to image data. Convolutional neural networks (CNN) were trained to learn the features of faults, which can be used for classification. They did not investigate the impact of potential sensor errors on the classification accuracy or the ability to discriminate between multiple simultaneous faults.

Kim et al. [16] developed a reference model of a residential heat pump system for fault detection and diagnosis. Although they did not simply lump the refrigerant cycle into a single component, they did not consider the physics of the system, but rather applied regression via multivariate polynomial regression (MPR) and artificial neural networks (ANN). In all cases, they used the outdoor dry-bulb temperature, indoor dry-bulb temperature, and indoor dew point temperature as independent inputs, and predicted the values of seven dependent features including the evaporating temperature, condensing temperature, and air temperature changes across the heat exchangers.

Vapor compression cycles such as chillers and heat pumps can be subject to a wide range of faults. Aguilera et al. [17] conducted a review of common faults in large-scale heat pumps which covered 53 case studies. A larger number of faults were identified in the compressor and evaporator or source heat exchanger, with very few in the condenser. Compressor faults include noise and vibration that cause pressure pulsation, defective shaft seals that cause leakage and additional frictional torque, defective capacity control valves resulting in additional thermodynamic losses, and defective electric motors. In the evaporators, the most frequent faults were fouling, which may affect the pressure drop and thermal performance, followed by corrosion, and excessive frosting which can lead to a reduction in performance and capacity. The most common faults in general were fouling in the heat exchangers, refrigerant leakage, noise and vibration, and defective control systems. Rogers et al. [18] and Li and Braun [19] also list the types of faults typically diagnosed in vapor compression air-conditioning systems. These include undercharge, overcharge, liquid-line restrictions, compressor valve leakage, condenser and evaporator fouling or reduced air flow, and non-condensables within the refrigerant lines. Piancento and Talamo [20] list typical faults, such as the fouling of the evaporator and condenser, refrigerant undercharge, compressor valve leakage, and extra superheating.

Parameter identification is one possible way of identifying faults when combined with a model of a physical system. Parameter identification techniques determine the

unknown values of specific parameters that characterize a physical system by minimizing the difference between the predicted and measured performance data via an optimization technique or a combination of techniques. These range from traditional Newton or quasi-Newton methods, least squares algorithms [21], genetic algorithms [22,23], metaheuristic optimization [24], and convolutional neural networks [25].

In this paper, we propose the use of a fundamental physics-based thermofluid model of a heat pump cycle combined with deep learning-based surrogate models and parameter identification for online condition monitoring. Whereas many existing condition-monitoring approaches focus on detection only, the proposed methodology allows the simultaneous detection, location, and quantification of degradation occurring in the different heat pump system components. This approach can discriminate between multiple simultaneous faults and has not been applied before to heat pumps or other vapor compression cycles. The methodology is demonstrated with the aid of synthetically generated data, which include the effect of measurement uncertainty.

2. Materials and Methods

This section first introduces the actual heat pump used as a case study, then the overall methodology that is proposed, followed by a description of the thermofluid model, the degradation factors employed, and then the surrogate models and parameter identification technique.

2.1. Case Study Heat Pump

The heat pump that is used as the case study is a deep mine air-cooling unit (ACU), of which a schematic is shown in Figure 1. It is utilized in mines at underground depths of more than 1000 m to cool the working environment at development ends where air temperatures are typically higher than 30 °C with high humidity. It cools the air down to around 22–23 °C, while rejecting heat to a cooling water stream that is provided from the surface at around 20–25 °C.

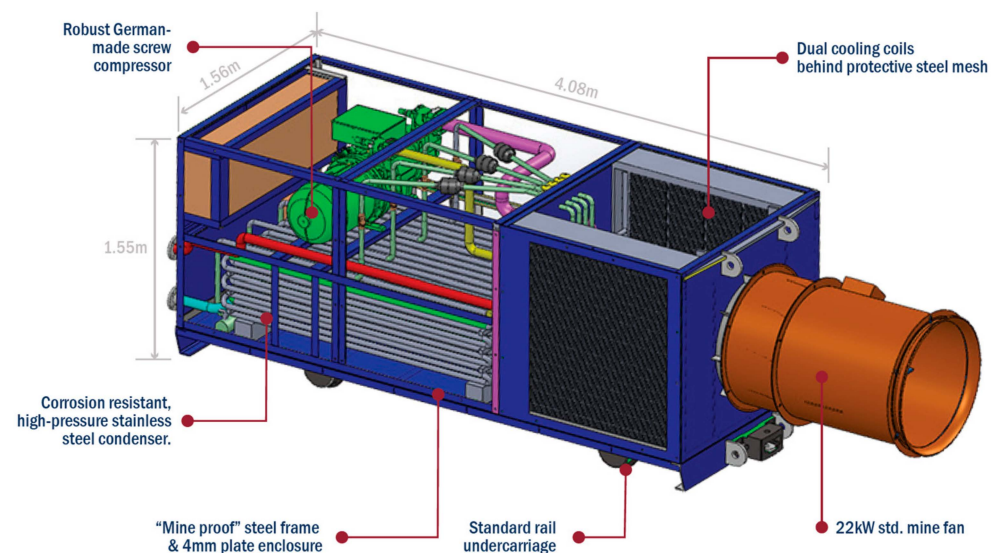


Figure 1. Image of the deep mine air-cooling unit used as a case study (courtesy of M-Tech Industrial (Pty) Ltd., Potchefstroom, South Africa).

The heat pump weighs around 4.5 tons and consists of a standard refrigeration cycle layout with a R407c screw compressor, a water-cooled tube-in-tube condenser, and a moist air finned tube evaporator coil. Specific details of the design and component specifications will not be provided here since it is commercially sensitive information.

If breakdowns occur, the coordination involved is complex and the costs can be substantial. Since the machine is lowered down the mine shaft to significant depths, it is

not feasible to extract it for regular maintenance, and maintenance and repairs occurring deep underground is challenging. Any breakdown can also result in significant financial losses due to the loss of production. An online condition-monitoring system that enables the pro-active detection, location, and quantification of degradation can therefore provide significant benefits in this specific case.

2.2. Methodology

The proposed condition-monitoring methodology is presented in the schematic shown in Figure 2. The discussion that follows will refer to the various blocks within the dashed lines that are labelled A to G.

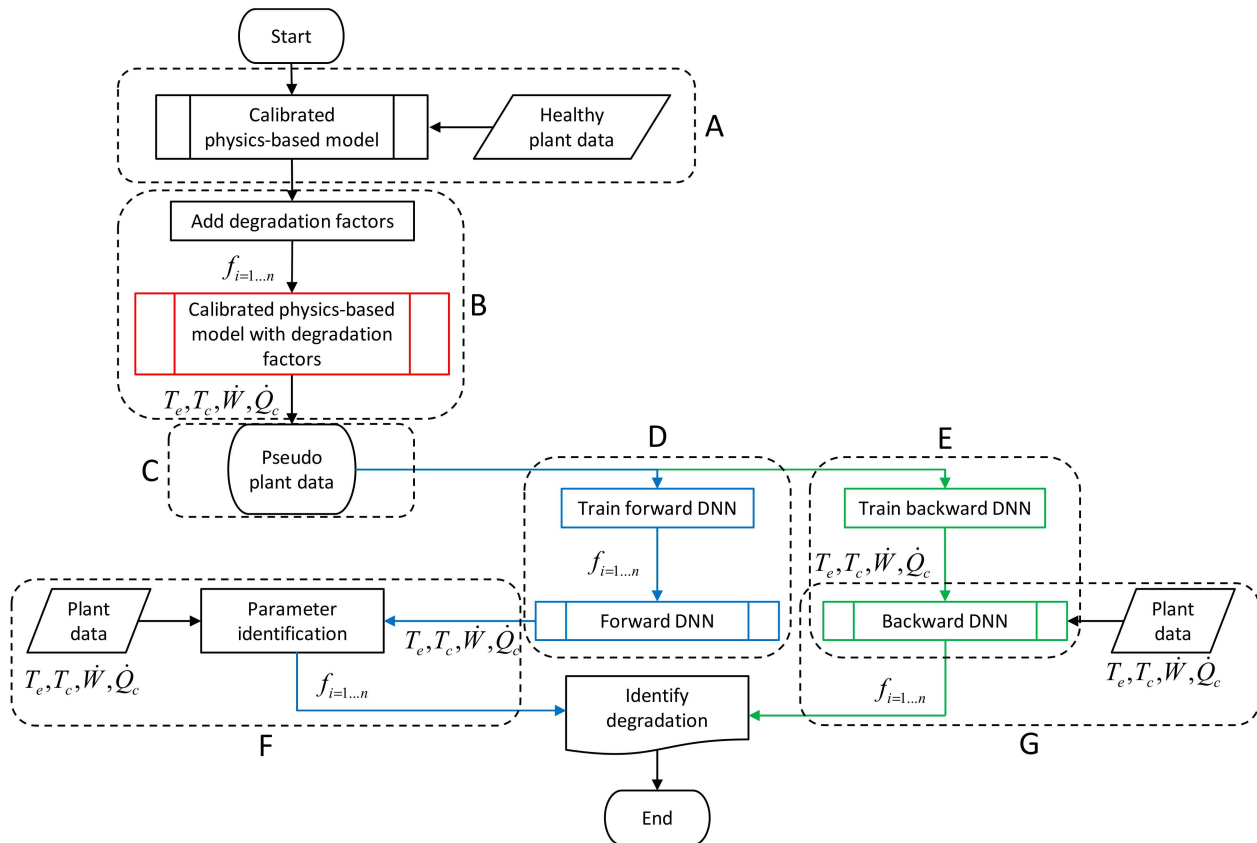


Figure 2. Flow diagram of the proposed condition-monitoring methodology.

Block A: An essential element is to have a representative and properly calibrated thermofluid model of the heat pump cycle. The model must be constructed so that it represents the characteristics of the individual components with sufficient detail to capture the effects of degradation on its performance. Furthermore, it must include the coupling amongst the different components since the performance of any one component is heavily dependent on the performance of the other components [20], and therefore influences the performance of the integrated cycle. Whatever model is employed must be calibrated with the aid of data from healthy plant measurements. Successful calibration is of prime importance, but will not be addressed here since the aim of this paper is to demonstrate the overall methodology rather than the complexity of the model calibration, which is a major topic. Furthermore, in the present work, the thermofluid model will be simplified when compared to reality, but still complex enough to demonstrate the thermofluid phenomena involved. More details of the thermofluid model will be provided in Section 2.3, and the following work assumes that the model is calibrated appropriately.

Block B: An array of degradation factors ($f_{i=1..n}$) that represent specific modes of degradation in the components can now be added to the various component models. This

implies that, by adjusting a specific degradation factor, it is possible to simulate the effect of the degradation on the specific component, as well as its impact on the performance of the complete integrated cycle. Each of these degradation factors are defined to have a value between zero and one. A value of one implies no degradation, i.e., a completely healthy component, while a value of zero implies complete degradation. It would of course be ideal if the actual degradation mechanisms could be integrated into the model, for instance, by physically simulating the flow leakage around seals or valves, or additional friction on moving parts due to wear. However, this would place an unnecessarily high burden on the simulation model, with an associated high burden for accurate calibration, because the final detail physical fault diagnosis must still be conducted on site by a competent technician. Since a simplified thermofluid model is employed here, not all degradation phenomena will be simulated, and only four degradation factors are included for demonstration purposes. More detail of the specific degradation factors included in this study will be provided in Section 2.4.

In this study, the performance of the heat pump cycle will be characterized by four readily measurable parameters, namely the evaporating temperature (T_e), the condensing temperature (T_c), the compressor power input requirement ($\dot{W}$), and the rate of heat transfer in the condenser ($\dot{Q}_c$). It is of course possible to employ more parameters, but the aim of the paper is to demonstrate that the methodology can be employed using a limited number of parameters that can readily be measured on the real plant. In the current work, the pseudo ‘measured’ values will be assumed to be the quasi-steady state mean values measured over a period, rather than instantaneous values. This approach is common [18] and implies a degree of data pre-processing, which will not be addressed here. However, the current work will investigate the impact of varying degrees of noise in the measured plant data. This will be addressed in more detail in Section 3.3.

Block C: The next step is to apply the model to generate pseudo plant operational data that cover the complete envelop of operation as well as all possible combinations of degradation factors. The operational envelope is defined by the ranges of the various boundary values that serve as inputs to the model. In this case, it will be the inlet dry-bulb temperature of the air into the evaporator heat exchanger (T_{ai}), the inlet relative humidity of the air (RH_i), the mass flow rate of water through the condenser heat exchanger ($\dot{m}_w$), and the inlet temperature of the water (T_{wi}). Note that for the purposes of this study, it is assumed that the mass flow rate of the air through the evaporator heat exchanger ($\dot{m}_a$) remains constant. This is a reasonable assumption since the plant is fitted with a single fixed-speed fan drawing air from the surroundings. The data generation process is based on a suitable design of experiments to ensure that the data cover the complete operational envelope together with all possible combinations of degradation factors. More details of the training data generation will be provided in Section 3.1.

Blocks D and E: Next, the plant data are used to train either a ‘forward’ DNN or a ‘backward’ DNN. The forward DNN takes as input features all the operational boundary values together with all the degradation factors, and predicts as output labels the values of the performance parameters, i.e., the parameters measured on the real plant. The backward DNN takes as input features all the boundary values together with the performance parameters, and predicts as output labels the values of all the degradation factors. More details of the surrogate models will be provided in Section 2.5.

Block F: The forward DNN is applied in the condition-monitoring process via comparing predicted values with that of online measurements taken on the real plant during operation. It is combined with a suitable parameter identification technique to discover the appropriate values of the degradation factors that will result in the best predictions of the performance parameters. In essence, the parameter identification technique solves an optimization problem with the degradation factors as the independent variables and the mean squared error between the predicted and measured performance parameters as the target function. More details of the parameter identification will be provided in Section 2.6.

Block G: The application of the backward DNN does not require a parameter identification technique. It simply takes as input the measured operational boundary values as well as the measured performance parameters, and directly predicts the values of the degradation factors.

The modeling and analyses for this study were performed on an Intel® Xeon® Gold 6346 CPU @ 3.10 GHz, with 36 MB cache, 64 GB RAM, and 16 cores. The models were developed in Python 3.12.2 64 bit, while making use of NumPy for general mathematical programming, SciPy for root finding and optimization, pandas for data manipulation and analysis, Matplotlib and Engineering Equation Solver (EES) for visualization, CoolProp for fluid property calculations, pyDOE for the design of experiments, as well as scikit-learn and PyTorch for machine learning.

2.3. Thermofluid Model of the Heat Pump

The thermofluid model of the heat pump consists of five parts, namely the refrigerant cycle model, the compressor model, the condenser heat exchanger model, the evaporator heat exchanger model, and finally the integration between the overall cycle and the component models.

2.3.1. Refrigerant Cycle Model

A schematic of the refrigerant cycle is shown in Figure 3, while the corresponding temperature-specific entropy diagram is shown in Figure 4.

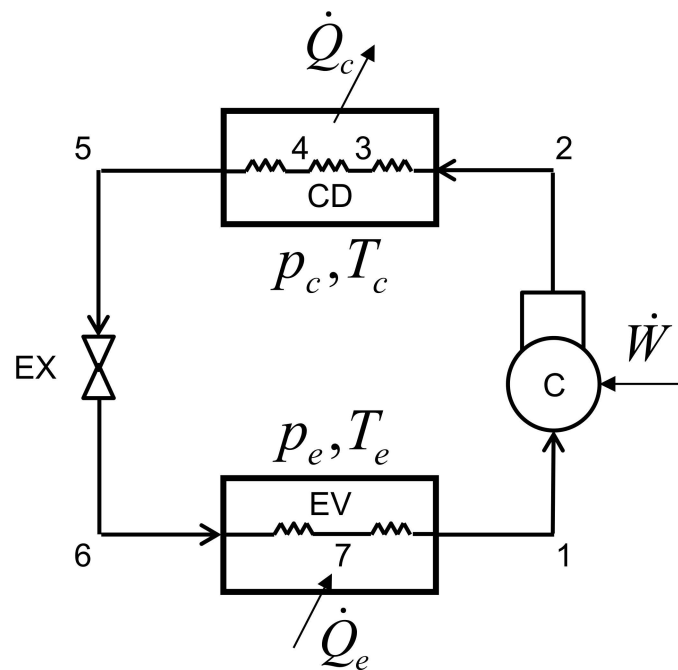


Figure 3. Schematic of the refrigerant cycle.

The primary components in the heat pump are the refrigerant compressor (C), condenser heat exchanger (CD), expansion valve (EX), and evaporator heat exchanger (EV). For the purposes of the current work, the presence of secondary components such as pipes, valves, dryers, accumulators, etc., is ignored. Heat losses to the surroundings and all pressure losses due to friction, inlets, outlets, bends, etc., are also neglected. This is performed to simplify the model since the aim of the paper is not to present a complete thermofluid model, but rather to demonstrate the condition-monitoring methodology. However, all the secondary components as well as heat and pressure losses could also readily be incorporated.

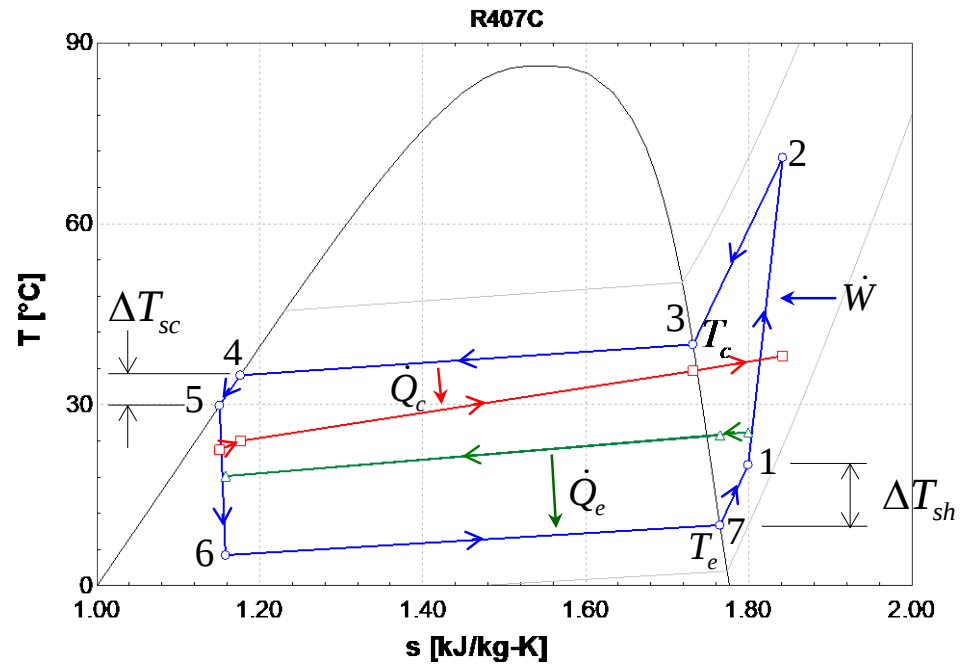


Figure 4. Refrigerant cycle (blue) depicted on the temperature versus entropy diagram as well as temperature traces of the water through the condenser (red) and air through the evaporator (green).

Given the simplified nature of the refrigerant cycle, it is analyzed via the conventional thermodynamic cycle analysis approach. Refrigerant enters the compressor as superheated vapor at point 1 and is compressed in an irreversible process to superheated vapor at a higher pressure and temperature at point 2. The required power input to the compressor is designated as $\dot{W}$, and the irreversibility of the process is represented by an appropriate value for the isentropic efficiency, defined as

$$\eta_s = \frac{\dot{W}_s}{\dot{W}} \quad (1)$$

where $\dot{W}_s$ is the isentropic (reversible and adiabatic) power input.

From the outlet of the compressor, the refrigerant enters the condenser where it rejects heat and is desuperheated to saturated vapor at point 3. The pressure and temperature at point 3 are designated as the condensing pressure p_c and condensing temperature T_c . It is then condensed to saturated liquid at point 4, and then cooled further to subcooled liquid at point 5. Therefore, the condenser contains superheated, two-phase, and subcooled regions, and the total rate of heat transfer in the condenser is designated as $\dot{Q}_c$.

From there, it is expanded through the expansion valve to a lower pressure while flashing to a mixture of saturated liquid and vapor at point 6. Here, it enters the evaporator heat exchanger where it absorbs heat while evaporating further to saturated vapor only at point 7. The pressure and temperature at point 7 is designated as the evaporating pressure p_e and evaporating temperature T_e . It is then superheated to point 1 at the inlet of the compressor. The evaporator therefore contains a two-phase region and a superheated region, and the total rate of heat transfer in the evaporator is designated as $\dot{Q}_e$.

Since pressure losses are ignored, the pressure at the inlet of the compressor is also equal to p_e , and the pressure at the outlet of the compressor equal to p_c .

The degree of subcooling is defined as

$$\Delta T_{sc} = T_4 - T_5 \quad (2)$$

and it is specified as a controlled parameter. Amongst the major components in the current model, the bulk of the refrigerant mass is located inside the subcooled region within the

condenser, since the internal volume of the condenser is large and there is an order of magnitude difference between the densities of the saturated liquid and saturated vapor. The assumption is therefore made that the degree of subcooling serves as a proxy for the refrigerant charge, i.e., the mass of refrigerant that is present in the closed cycle. If less refrigerant is charged, the degree of subcooling will be less, while more refrigerant charge will result in more subcooling. A more sophisticated feature for expressing refrigerant charge was developed by Li and Braun [19], which could also be incorporated as part of the proposed methodology, but will not be included in the current work.

The degree of superheating is defined as

$$\Delta T_{sh} = T_1 - T_7 \quad (3)$$

and is also specified as a controlled parameter since it is typically controlled via the expansion valve. In this simplified model, the degree of superheating therefore serves as a proxy for a detailed component model of the expansion valve.

The energy balance for the refrigerant through the condenser, while neglecting potential and kinetic energy difference between the inlet and outlet, is given by

$$\dot{Q}_c = \dot{m}_r(h_2 - h_5) \quad (4)$$

with $\dot{m}_r$ being the refrigerant mass flow rate, and h being the specific enthalpy at the respective points. The energy balance for the refrigerant through the evaporator is given by

$$\dot{Q}_e = \dot{m}_r(h_1 - h_6) \quad (5)$$

The energy balance through the compressor is given by

$$\dot{W} = \dot{m}_r(h_2 - h_1) \quad (6)$$

The flow through the expansion valve is assumed to be adiabatic and therefore

$$h_6 = h_5 \quad (7)$$

The refrigerant cycle takes as input the values of ΔT_{sc} , ΔT_{sh} , p_e , and p_c . From p_e and ΔT_{sh} , it determines all the properties at point 1. Using p_e and p_c , it then obtains the value of $\dot{m}_r$ and η_s from the compressor model (described in Section 2.3.3) and calculates the properties at point 2 based on Equations (1) and (6). It then goes on to calculate the properties at points 3, 4, and 5, also using ΔT_{sc} , and then determines $\dot{Q}_c$ from Equation (4). Finally, it finds the properties at point 6 from Equation (7), followed by point 7, and then determines $\dot{Q}_e$ from Equation (5). The outputs of the refrigerant cycle model are therefore the values of the $p_{1..7}$, $h_{1..7}$, and $T_{1..7}$ (including T_e and T_c), together with $\dot{m}_r$, $\dot{W}$, $\dot{Q}_c$, and $\dot{Q}_e$.

2.3.2. Fluid Property Calculations

The R407c and water properties are calculated using the PropsSI function, while the moist air properties are calculated using HAPropsSI. However, it was found that in the execution of the models, a large fraction of the total time is required to calculate the fluid properties. Furthermore, it was found that within iterations, it is possible for the PropsSI function to be called outside of its envelope of application. To remedy this, a custom suite of modules was developed that use the original PropsSI functions to generate fluid property lookup tables, and then use these tables to find the required R407c and water properties where necessary.

The lookup tables were generated using the original PropsSI functions. For R407c, a rectangular data grid of 400 equally spaced pressure values by 400 equally spaced enthalpy values was used, i.e., 160,000 data points per table. These pressure values range from 200 kPa to 5000 kPa, and the enthalpy values from 150 kJ/kg to 500 kJ/kg. For water, a 500 by 500 grid was used, with the pressure values ranging from 5 kPa to 25,000 kPa and the enthalpy values from 50 kJ/kg to 4000 kJ/kg. Separate lookup tables were generated for temperature, specific entropy, density, thermal conductivity, and dynamic viscosity as a

function of pressure and enthalpy. Furthermore, tables were generated for the values of saturated liquid enthalpy, temperature, specific entropy, density, thermal conductivity, and dynamic viscosity as a function of the saturation pressure, as well as the corresponding saturated vapor properties. The lookup tables are read by another custom module that uses the SciPy class `scipy.interpolate.RectBivariateSpline` to determine the fluid property values. All the lookup tables and procedures were thoroughly validated through comparing the interpolated values from the tables with values derived from the original PropsSI function.

The two major benefits of the lookup table approach are that it is significantly faster than the original functions, and that it automatically bounds the values of the pressure and enthalpy inputs to the envelope covered in the lookup table. This avoids unrealistic fluid property calls within iterations. A speedup of approximately 100 times is realized for determining the temperature from the pressure and enthalpy, and approximately 200 times for determining the specific entropy from the pressure and enthalpy.

2.3.3. Compressor Model

The compressor model applied here is the same as that of Jahnig et al. [26]. The refrigerant mass flow rate is given by

$$\dot{m}_r = \frac{V_s}{v_i} N \left(1 - VdVs \left(\left(\frac{p_c}{p_e(1 - \delta p)} \right)^{1/\gamma} - 1 \right) \right) \quad (8)$$

where V_s is the swept volume, v_i is the specific volume of the refrigerant gas at the inlet, N is the rotational speed in Hz, $VdVs$ is the effective clearance volume ratio, δp is the effective pressure drop at the inlet, and γ is the ratio of the specific heats of the refrigerant gas at the inlet.

The isentropic efficiency of the compressor is a function of the inlet and outlet pressures, and is given by the bivariate cubic polynomial as follows:

$$\eta_s = c_0 + c_1 p_e + c_2 p_c + c_3 p_e^2 + c_4 p_e p_c + c_5 p_c^2 + c_6 p_e^2 p_c + c_7 p_e p_c^2 + c_8 p_c^3 \quad (9)$$

where $c_{0..8}$ are the appropriate correlation coefficients. (Note that the p_e^3 term was omitted on purpose to avoid potential overfitting in this case.) The value of V_s is typically provided by the OEM (original equipment manufacturer) of the compressor, while the values of $VdVs$, δp , and $c_{0..8}$ are determined via least squares curve fitting to the performance data provided by the OEM.

2.3.4. Condenser Heat Exchanger Model

The fact that the degree of subcooling serves as a proxy for the refrigerant charge necessitates a high level of detail to be resolved in the condenser model. It requires the accurate determination of the transition point between the two-phase flow region and the subcooled region, as well as between the superheated region and the two-phase region. The positions of the transition points are not known a priori and must be determined as part of the solution of the model. Therefore, in this work, the condenser heat exchanger is modeled in detail via a thermofluid network model [27]. A schematic of the model is shown in Figure 5.

In the thermofluid network model, the flow paths on both the refrigerant and water sides are discretized into n control volumes that are connected to represent the layout of the actual flow path in the heat exchanger. In this case, the flow configuration is purely counterflow, as shown in Figure 5. Besides the geometrical parameters, the input boundary values to the model are the refrigerant mass flow rate ($\dot{m}_r$) and overall inlet pressure (p_{ri}) and enthalpy (h_{ri}), as well as the water mass flow rate ($\dot{m}_w$) and overall inlet pressure (p_{wi}) and enthalpy (h_{wi}). Each control volume j is treated as a counterflow heat exchanger element for which the rate of heat transfer ($\dot{Q}_{c,j}$) is determined.

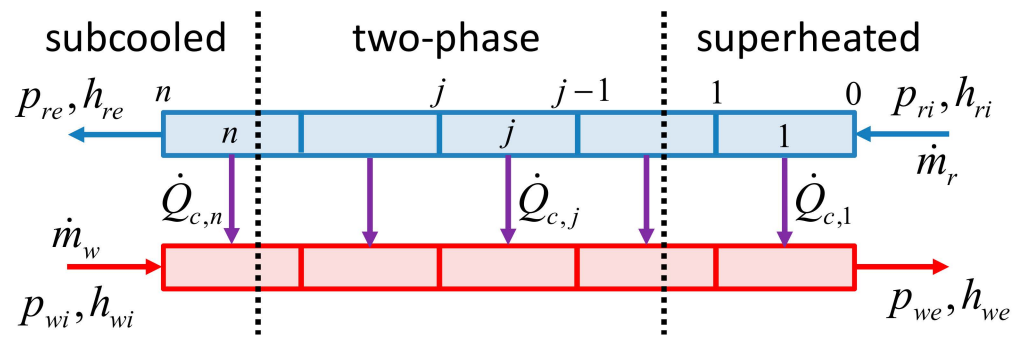


Figure 5. Schematic of the condenser heat exchanger model layout (not showing the actual number of increments).

By writing the energy balance for each refrigerant-side control volume, we obtain a set of $n + 1$ unknowns, namely $h_{r,0}$ to $h_{r,n}$, and $n + 1$ simultaneous equations given by

$$\begin{aligned}
 \dot{m}_r h_{r,0} &= \dot{m}_r h_{ri} \\
 -\dot{m}_r h_{r,0} + \dot{m}_r h_{r,1} &= \dot{Q}_{c,1} \\
 \dots &= \dots \\
 -\dot{m}_r h_{r,j-1} + \dot{m}_r h_{r,j} &= \dot{Q}_{c,j} \\
 \dots &= \dots \\
 -\dot{m}_r h_{r,n-1} + \dot{m}_r h_{r,n} &= \dot{Q}_{c,n}
 \end{aligned} \quad (10)$$

The right-hand side of Equation (10) contains the inlet boundary value and the source terms, i.e., the rate of heat transfer in each of the control volumes. A similar set of $n + 1$ unknowns, namely $h_{w,0}$ to $h_{w,n}$, and $n + 1$ simultaneous equations can be written for the water-side control volumes, namely

$$\begin{aligned}
 \dot{m}_w h_{w,0} - \dot{m}_w h_{w,1} &= \dot{Q}_{c,1} \\
 \dots &= \dots \\
 \dot{m}_w h_{w,j-1} - \dot{m}_w h_{w,j} &= \dot{Q}_{c,j} \\
 \dots &= \dots \\
 \dot{m}_w h_{w,n-1} - \dot{m}_w h_{w,n} &= \dot{Q}_{c,n} \\
 \dot{m}_r h_{r,n} &= \dot{m}_w h_{wi}
 \end{aligned} \quad (11)$$

The condenser model starts off with guessed values for the refrigerant and water pressures and enthalpies at the inlet and outlet of each control volume, from which the temperatures are determined using the appropriate fluid property relationships. The average values of pressure and enthalpy are then determined for each control volume, and from this, the average values of all the other fluid properties are determined. Based on the inlet, outlet, and average values of the fluid properties, the source term values are determined through calculating the rate of heat transfer in each of the control volumes. Following this, Equations (10) and (11) are solved using a linear equation solver function to update the enthalpy values, which in this case is `numpy.linalg.solve`. This process is repeated iteratively while applying relaxation to the calculated source term values until sufficient convergence is achieved.

The rate of heat transfer in each control volume is calculated using the well-known effectiveness-NTU method, where the maximum possible rate of heat transfer is calculated as

$$\dot{Q}_{cmax,j} = C_{min,j}(T_{r,j-1} - T_{w,j}) \quad (12)$$

and the actual rate of heat transfer as

$$\dot{Q}_{c,j} = \varepsilon_j \dot{Q}_{cmax,j} \quad (13)$$

where ε_j is the effectiveness and $C_{\min,j}$ is the minimum heat capacity of the control volume j . The minimum and maximum heat capacities are determined as

$$\begin{aligned} C_{\min,j} &= \min(\dot{m}_r c_{pr,j}, \dot{m}_w c_{pw,j}) \\ C_{\max,j} &= \max(\dot{m}_r c_{pr,j}, \dot{m}_w c_{pw,j}) \end{aligned} \quad (14)$$

where $c_{pr,j}$ and $c_{pw,j}$ are the specific heat capacities of the refrigerant and water, respectively, in the control volume j . For the pure counterflow layout, the effectiveness is then determined from

$$\varepsilon_j = \frac{1 - e^{-NTU_j(1-C_{r,j})}}{1 - C_{r,j}e^{-NTU_j(1-C_{r,j})}} \quad (15)$$

where

$$C_{r,j} = \frac{C_{\min,j}}{C_{\max,j}} \quad (16)$$

and

$$NTU_j = \frac{UA_j}{C_{\min,j}} \quad (17)$$

UA_j is the effective product of the overall heat transfer coefficient and the heat transfer surface area, which we will refer to as the UA value.

By neglecting the thermal resistance of the heat exchanger tube material and any fouling that may occur (potential fouling will be addressed via a degradation factor), the UA value can be determined from

$$\frac{1}{UA_j} = \frac{1}{h_{o,j}A_{ho,j}} + \frac{1}{h_{i,j}A_{hi,j}} \quad (18)$$

where $h_{o,j}$ and $h_{i,j}$ are the convective heat transfer coefficients on the refrigerant and water sides, respectively, and $A_{ho,j}$ and $A_{hi,j}$ are the outside and inside heat transfer areas, respectively, for the control volume j . The heat transfer coefficients for the refrigerant and water sides are determined in detail as described by Rousseau et al. [28]. This includes the values for the single-phase and two-phase regions on the refrigerant side within the heat exchanger.

On the refrigerant side, the single-phase heat transfer coefficient correlations are employed for control volumes situated completely within the single-phase region, while the two-phase correlations are employed for control volumes situated completely within the two-phase region. For control volumes containing the transitions from superheated to two-phase conditions and from two-phase to subcooled conditions, the weighting of the respective heat transfer areas are employed. However, the correct ratio between the single-phase and two-phase areas is not known a priori and forms part of the solution.

Figure 6 shows a schematic of a control volume on the refrigerant side that contains the transition from superheated to two-phase conditions. The appropriate ratio between the heat transfer areas can be derived from considering an energy balance for each region as follows:

$$\begin{aligned} \dot{m}_r(h_{ri} - h_{rg}) &= h_{sh}A_{h,sh}\Delta\bar{T} \\ \dot{m}_r(h_{rg} - h_{re}) &= h_{tp}A_{h,tp}\Delta\bar{T} \end{aligned} \quad (19)$$

where h_{ri} , h_{re} , and h_{rg} are the inlet, outlet, and saturated vapor enthalpies, respectively, h_{sh} and h_{tp} are the superheated region and two-phase region heat transfer coefficients, respectively, and $A_{h,sh}$ and $A_{h,tp}$ are the superheated region and two-phase region heat transfer areas, respectively. $\Delta\bar{T}$ is the difference between the average temperatures on the refrigerant and water sides that drives the rate of heat transfer. From this, we may write the heat transfer area ratio as

$$\frac{A_{h,sh}}{A_{h,tp}} = \frac{h_{ri} - h_{rg}}{h_{rg} - h_{re}} \frac{h_{tp}}{h_{sh}} \quad (20)$$

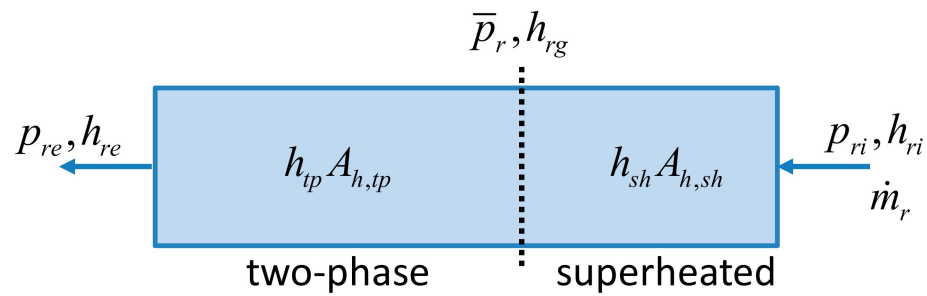


Figure 6. Schematic of a control volume in the refrigerant side that contains the transition from superheated to two-phase conditions.

Therefore,

$$\begin{aligned} A_{h,tp} &= A_h \left(\frac{h_{ri} - h_{rg}}{h_{rg} - h_{re}} \frac{h_{tp}}{h_{sh}} + 1 \right)^{-1} \\ A_{h,sh} &= A_h - A_{h,tp} \end{aligned} \quad (21)$$

where A_h is the total heat transfer area on the refrigerant side of the control volume. Note that as the values of the parameters in the brackets on the right-hand side of Equation (21) are adjusted between iterations, so will the heat transfer areas. It is also possible for the transition point to shift from one control volume to another in between iterations, and therefore the position of the transition point forms part of the final converged solution. A similar approach is applied for the areas around the transition from the two-phase region to the subcooled region.

Since a discretized network approach is used, it is necessary to ensure that the final solution is grid size-independent. A grid independence study was conducted as part of the model development process, and it was found that 50 increments provide a grid-independent solution over a wide range of operating conditions. Figure 7 shows the typical temperature distributions obtained at the nominal operating point with different grid sizes from $n = 5$ to $n = 50$, together with the resultant calculated rate of heat transfer $\dot{Q}_c$.

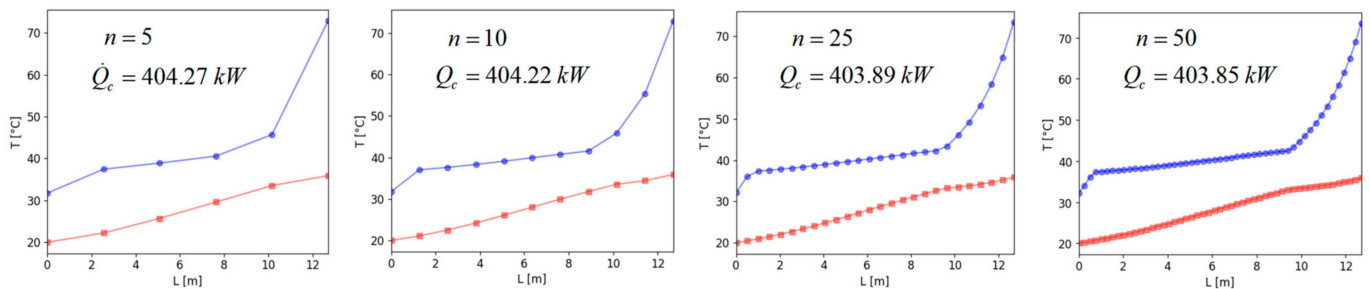


Figure 7. Calculated temperature distributions on the refrigerant side (blue) and water side (red) as a function of the normalized position through the condenser heat exchanger at nominal operating conditions for different grid sizes.

2.3.5. Evaporator Heat Exchanger Model

The evaporator model is also based on a thermofluid network approach. However, in this case, the physical layout can be approximated as a network of crossflow heat exchanger elements connected in a general counterflow configuration, as shown in the schematic in Figure 8.

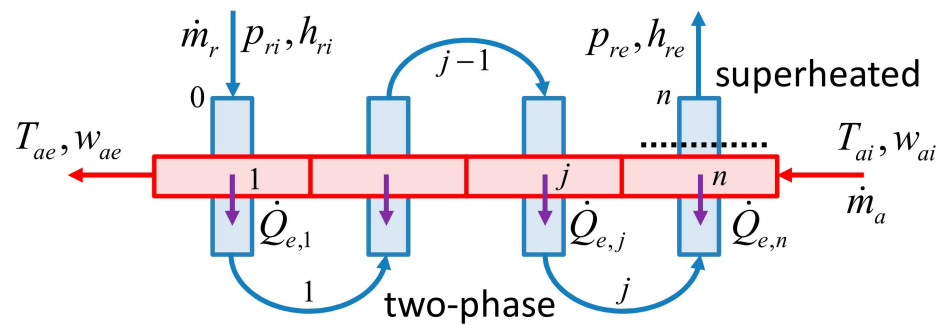


Figure 8. Schematic of the evaporator heat exchanger model layout (the actual number of increments is discussed in the main text).

On the refrigerant side, the same principles can be applied as in the condenser model in terms of the single- and two-phase regions and the transition between them. However, on the moist air side, the process involves combined heat and mass transfer rather than heat transfer only. If the temperature of the cold outside surface is below the dew point temperature of the moist air entering the control volume, the air will start to dehumidify with the condensation of vapor taking place on the surface. To account for the moisture content in the air, the boundary values on the air side are the dry-bulb temperature (T_{ai}) and absolute humidity (w_{ai}) in addition to the air mass flow rate ($\dot{m}_a$). The evaporator heat exchanger therefore exhibits two-phase flow on the refrigerant side, together with combined heat and mass transfer on the air side. Although it is quite feasible to construct the required detailed analysis, it is considerably more complicated than for heat transfer only.

As mentioned before, in the current study, the mass flow rate of air through the evaporator is assumed to be constant. Furthermore, the thermal resistance between the refrigerant and the air is dominated by the air side convective thermal resistance. This means that the overall UA value of the evaporator heat exchanger will not vary significantly. Given this, and to simplify the model for the purposes of the current study, it is assumed that the overall UA value of the evaporator heat exchanger is constant. It is therefore not necessary to perform a detailed analysis of the various convective heat transfer coefficients in order to continually update the UA value as the operating conditions change.

Given that the overall UA value is assumed to be constant and will not be calculated in detail, a further simplifying assumption can be made. Rather than discretizing the heat exchanger surface into multiple increments, it is only divided into two control volumes, namely one for the two-phase region and one for the superheated region. If at least these two different regions are not accounted for, it could lead to unrealistic results since the refrigerant and air temperatures may cross over in cases where the inlet temperatures are close together. Another point to note is that since only two control volumes are used, each of these can encapsulate several of the crossflow elements, which are all connected in an overall counterflow arrangement. Therefore, the flow configuration of the heat exchanger elements for each of the two control volumes will be assumed to be counterflow rather than crossflow.

Since the thermal resistance between the refrigerant and the air is dominated by the air-side convective thermal resistance, the outside metal temperatures in contact with the air can be approximated by the refrigerant-side temperatures. The rate of heat transfer is again calculated using the effectiveness-NTU approach, but modified to accommodate the mass transfer. Starting with guessed values for the air enthalpies and absolute humidities at the inlet and outlet of each control volume, the maximum possible rate of heat transfer is calculated on the refrigerant side and on the air side, and the minimum of the two is taken as the maximum possible value for the control volume. The effectiveness is then determined using the formulation for the pure counterflow configuration, and the total rate of heat transfer is determined. However, the energy balance on the air side must account

for the rate of condensation and the removal of the condensate from the control volume. The rate of heat transfer from the moist air to the refrigerant is written as

$$\dot{Q}_{e,j} = \dot{m}_a \left[(h_{a,j} - h_{a,j-1}) - (w_{a,j} - w_{a,j-1}) h_{f,j-1} \right] \quad (22)$$

where $h_{a,j}$ and $h_{a,j-1}$ are the moist air enthalpies per kilogram of dry air at points j and $j - 1$, respectively, $w_{a,j}$ and $w_{a,j-1}$ are the moist air absolute humidity per kilogram of dry air at points j and $j - 1$, respectively, and $h_{f,j-1}$ is the enthalpy of saturated liquid at the metal surface temperature at point $j - 1$. The rate of condensation is given by

$$\dot{m}_w = \dot{m}_a (w_{a,j} - w_{a,j-1}) \quad (23)$$

To determine the absolute humidity at the outlet of each control volume, we apply the principle of a contact factor [29] or bypass factor [19]. It essentially assumes that the dehumidification process traces a straight line on the enthalpy versus absolute humidity plain towards the saturation conditions at the metal surface temperature. Therefore,

$$w_{a,j-1} = w_{a,j} - (w_{a,j} - w_{sat,j-1}) \left(\frac{h_{a,j} - h_{a,j-1}}{h_{a,j} - h_{sat,j-1}} \right) \quad \text{if } w_{sat,j-1} < w_{a,j} \\ w_{a,j-1} = w_{a,j} \quad \text{if } w_{sat,j-1} \geq w_{a,j} \quad (24)$$

where $w_{sat,j-1}$ and $h_{sat,j-1}$ are the absolute humidity and enthalpy, respectively, of saturated air at the metal surface temperature at point $j - 1$. The conditional statements for the absolute humidity, shown in Equation (24), imply that there will be no dehumidification if the metal surface temperature is above the dew point temperature of the air at the inlet.

Despite the simplification achieved with only two control volumes, the division of the overall UA value between the two-phase and superheated regions is not known a priori and must be determined as part of the overall solution. We define the evaporator superheated area fraction f_{EVSH} as the fraction of the overall UA value that is in the superheated region, so that

$$UA_{e,SH} = f_{EVSH} UA_e \\ UA_{e,TP} = (1 - f_{EVSH}) UA_e \quad (25)$$

UA_e is the constant overall UA value of the evaporator heat exchanger, and $UA_{e,SH}$ and $UA_{e,TP}$ are the UA values of the superheated and two-phase regions, respectively. This implies that the appropriate value of f_{EVSH} must be determined as part of the overall solution.

Determining the distribution of the overall UA value amongst the different regions is conducted differently in the evaporator than in the condenser. In the condenser model, the energy balance equations for the refrigerant side and the water side (Equations (10) and (11)) are solved iteratively until convergence, where the rate of heat transfer between the two fluids is balanced for each incremental control volume. In the evaporator model, the refrigerant states (points 6, 7, and 1 in Figure 4) are provided by the refrigerant cycle model as fixed input states, while the value of f_{EVSH} is provided by the overall integration module (which will be described in Section 2.3.6). The evaporator model then determines the resultant rate of heat transfer from the moist air to the refrigerant in the two-phase and the superheated regions, respectively, and passes these values back to the overall integration module. The overall integration module then ensures that the rates of heat transfer between the refrigerant cycle and the air in the evaporator are balanced via an overarching iterative process.

Figure 9 shows the calculated temperature distributions on the refrigerant and air side as a function of the normalized position in the evaporator, as well as the combined cooling and dehumidifying process on the psychrometric chart at the nominal operating conditions.

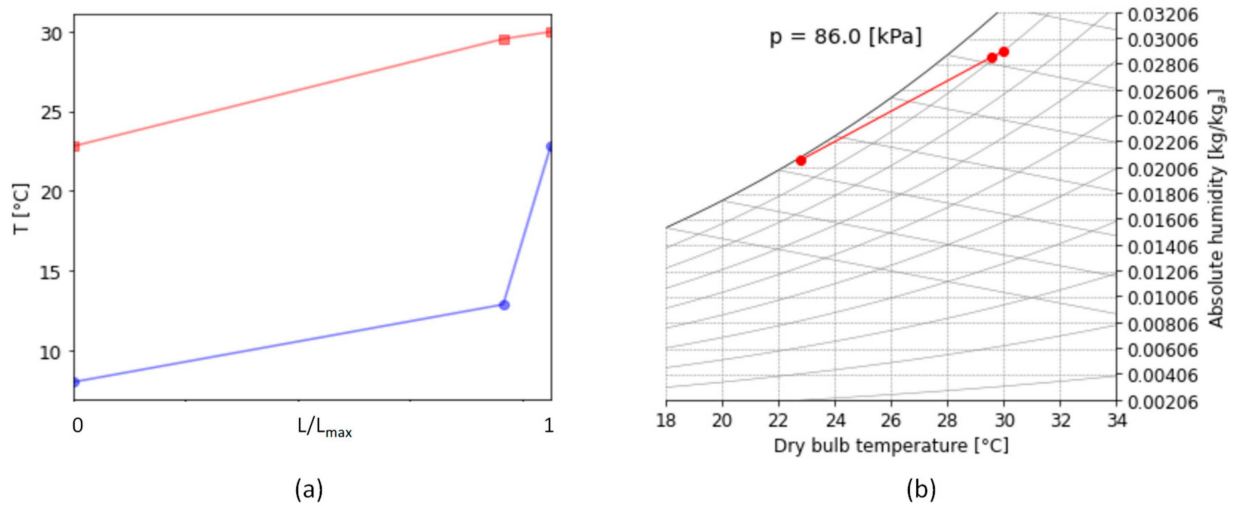


Figure 9. (a) Calculated temperature distributions on the refrigerant side (blue circles) and air side (red squares) as a function of the normalized position and (b) cooling and dehumidifying process on the psychrometric chart through the evaporator heat exchanger.

2.3.6. Overall Integration of the Refrigerant Cycle and Component Models

The module that is responsible for the overall integration between the refrigerant cycle and the respective component models starts with guessed values for the evaporating pressure, the condensing pressure, and the evaporator superheated area fraction. It then employs the non-linear root-finding function `scipy.optimize.fsolve` to solve for the appropriate values of p_e , p_c , and f_{EVSH} that form the roots of the following three functions:

$$\begin{aligned} f_1(p_e, p_c, f_{EVSH}) &= \left| \frac{\dot{Q}_{c,r} - \dot{Q}_{c,w}}{\dot{Q}_{c,r}} \right| \\ f_2(p_e, p_c, f_{EVSH}) &= \left| \frac{\dot{Q}_{e,r} - \dot{Q}_{e,a}}{\dot{Q}_{e,r}} \right| \\ f_3(p_e, p_c, f_{EVSH}) &= \left| \frac{\dot{Q}_{eSH,r} - \dot{Q}_{eSH,a}}{\dot{Q}_{eSH,r}} \right| \end{aligned} \quad (26)$$

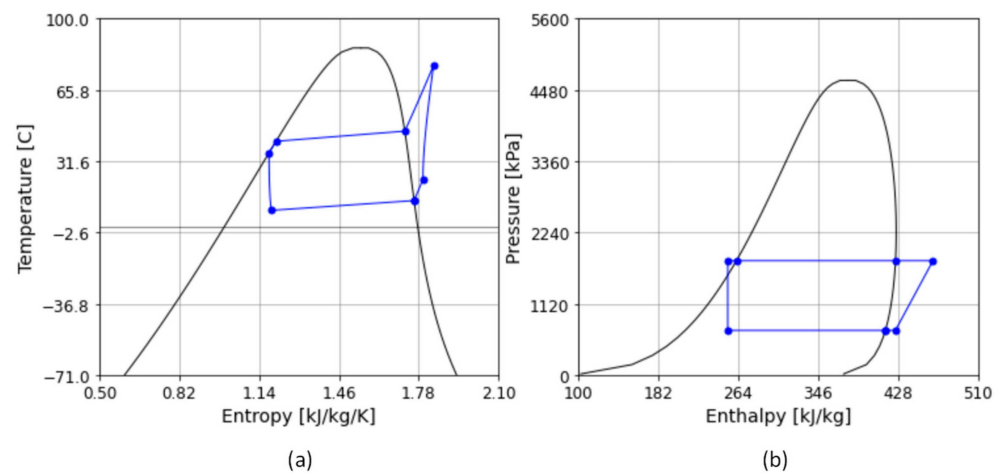
$\dot{Q}_{c,r}$, $\dot{Q}_{e,r}$, and $\dot{Q}_{eSH,r}$ are the total rate of heat transfer in the condenser, the total rate of heat transfer in the evaporator, and the rate of heat transfer in the superheated region of the evaporator, respectively, as calculated in the refrigerant cycle model combined with the compressor model. $\dot{Q}_{c,w}$ is the total rate of heat transfer in the condenser as calculated in the condenser heat exchanger model, while $\dot{Q}_{e,a}$ and $\dot{Q}_{eSH,a}$ are the total rate of heat transfer from the air and the rate of heat transfer from the air in the superheated region, as calculated in the evaporator heat exchanger model. In essence, this module adjusts the values of p_e , p_c , and f_{EVSH} to balance the rates of heat transfer calculated via the combined refrigerant cycle and compressor model and the heat exchanger models. The thermofluid model solution time is typically 8.2 s.

The nominal operating conditions of the heat pump as predicted via the thermofluid model are presented in Table 1.

Table 1. Nominal operating conditions of the heat pump predicted via the thermofluid model.

Compressor		Condenser		Evaporator		Cycle	
$\dot{m}_r$ [kg/s]	1.889	$\dot{m}_w$ [kg/s]	6.090	$\dot{m}_a$ [kg/s]	11.4	ΔT_{sc} [°C]	5.4
p_e [kPa]	696.2	p_w [kPa]	386.0	p_a [kPa]	86.0	ΔT_{sh} [°C]	10.0
p_c [kPa]	1653.2	T_{wi} [°C]	20.0	T_{ai} [°C]	30.0	T_e [°C]	12.4
$\dot{W}$ [kW]	68.5	T_{we} [°C]	35.9	T_{ae} [°C]	22.7	T_c [°C]	42.7
η_s	0.607	$\dot{Q}_c$ [kW]	403.8	RH_i	0.90	$COP_{cooling}$	4.9
				RH_e	0.99		
				$\dot{Q}_e$ [kW]	335.3		

Figure 10 shows the calculated values of the temperature vs. specific entropy and pressure vs. specific enthalpy at the nominal operating conditions.

**Figure 10.** (a) Temperature vs. specific entropy and (b) pressure vs. specific enthalpy diagrams calculated with the thermofluid model at the nominal operating conditions.

2.4. Degradation Factors

The model described above allows the analysis of numerous degradation mechanisms. However, as stated before, only four degradation factors are included in the current work to demonstrate the proposed methodology. These degradation factors are for the compressor mass flow rate (f_m), the compressor efficiency (f_η), the evaporator heat transfer (f_{ev}), and the condenser heat transfer (f_{cd}). Each of these can be applied to a specific area in the thermofluid model and represents one or more specific modes of degradation on the physical machine.

The compressor mass flow rate degradation factor is introduced in the model via multiplying it with the mass flow rate calculated via the compressor model. The “degraded” refrigerant mass flow rate is therefore simply

$$\dot{m}_r = f_m \dot{m}_r^* \quad (27)$$

where $\dot{m}_r^*$ is determined from Equation (8). The physical degradation represented by f_m is anything that results in the compressor delivering less than the expected mass flow rate at a given operating point defined by p_e and p_c . This could for instance include leakage around the seals or valves, which may short-cycle the discharge gasses, resulting in lower-than-expected mass flow rates throughout the whole system. The compressor displacement and clearance volume ratio are assumed to be constant since they are typically not affected by degradation [19].

Similarly, the compressor efficiency degradation factor is introduced in the model by multiplying it with the efficiency calculated via the compressor model. The “degraded” compressor efficiency is therefore simply

$$\eta_s = f_\eta \eta_s^* \quad (28)$$

where η_s^* is determined from Equation (9). The physical degradation represented by f_η is anything that results in a larger power input requirement than expected at a given operating point defined by p_e and p_c . The physical reason for this could, for instance, be additional friction on the moving part due to wear or an electrical fault in the motor.

The evaporator heat transfer degradation factor is multiplied with the overall UA value in the evaporator model. The “degraded” overall UA value is therefore

$$UA_e = f_{ev} UA_e^* \quad (29)$$

The physical degradation here is anything that impedes the rate of heat transfer between the refrigerant and the moist air in the evaporator. This could include fouling inside the tubes or on the outside of the tubes or fins. As stated earlier, the healthy overall UA value of the evaporator is assumed to be constant.

In the condenser model, the overall UA value is calculated in detail within each incremental control volume, as defined in Equation (18). The condenser heat transfer degradation factor is applied by multiplying it with the calculated UA value of each incremental control volume. The “degraded” overall UA value is therefore

$$UA_c = f_{cd} UA_c^* \quad (30)$$

Like the evaporator, the physical degradation here is anything that impedes the rate of heat transfer between the refrigerant and the water inside the condenser. This could include fouling inside the inner fluted tube on the water side or on the outside of the fluted tube on the refrigerant side. Interestingly, it would be possible to apply multiple condenser heat exchanger degradation factors to different areas within the condenser layout, thereby potentially identifying the location of the degradation more precisely. However, this is not explored within the scope of the current work.

2.5. Surrogate Models

Although the calculation time of the thermofluid model is only around eight seconds, it is not quick enough to employ directly in the proposed methodology where hundreds or thousands of simulations are needed to complete the parameter identification process. Therefore, very fast and accurate surrogate models are required. The surrogate models selected for this purpose are deep learning-based standard MLP (Multi-Layer Perceptron) neural networks. MLPs are well-suited to supervised learning where input features are mapped to output labels through tuning a set of weights and biases in order to minimize a selected cost function.

As shown schematically in Figure 11, the MLP networks consist of an input layer, one or more hidden layers with several fully connected neurons per layer, and an output layer. The vector of input features $\bar{x}$ serves as an input for the first hidden layer, while the final predicted output layer vector is $\hat{y}$. For each hidden layer l , the output is calculated in two steps. The first step is given by

$$\bar{z}_l = \bar{h}_{l-1} \bar{w}_l + \bar{b}_l \quad (31)$$

where $\bar{h}_{l-1}$ is the output signal vector from the previous layer, $\bar{w}_l$ are the weight matrices, and $\bar{b}_l$ are the bias vectors. $\bar{z}_l$ is then passed through an activation function σ

$$\bar{h}_l = \sigma(\bar{z}_l) \quad (32)$$

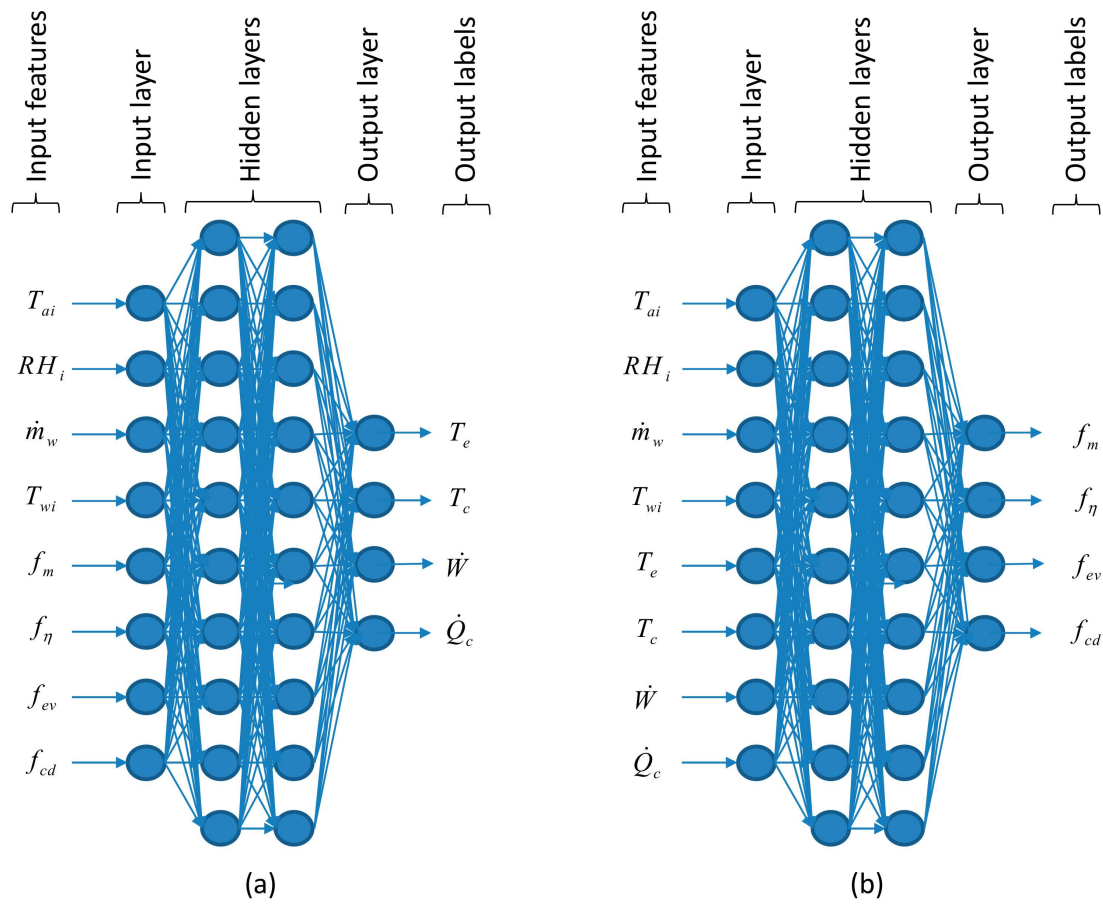


Figure 11. Schematic of the forward (a) and backward (b) MLP surrogate models. (The actual number of layers and neurons are discussed in the main text).

For the hidden layers, the activation function employed here is the hyperbolic tangent [30], while, for the output layer, it is linear, which is typical for regression-type problems.

The cost function employed here is the mean squared error (MSE) between the known output labels $\bar{y}$ and those predicted by the MLP $\hat{y}$, defined as

$$J_{MSE} = \frac{1}{n} \sum_{j=1}^n (y_i - \hat{y}_i)^2 \quad (33)$$

The weights and biases are tuned in an iterative process, starting with the forward propagation via (31) and (32), and followed by a backward propagation. The gradients of the cost function with respect to the weights and biases are first determined, and then the weights and biases are updated using the gradient descent algorithm. In the current work, the Adam [30] algorithm is employed for this purpose.

The input features for the forward surrogate model are the operational boundary values, namely the evaporator air inlet temperature T_{ai} and relative humidity RH_i and the condenser water mass flow rate $\dot{m}_w$ and inlet temperature T_{wi} , together with the values of the four degradation factors f_m , f_η , f_{ev} , and f_{cd} . The output labels are the plant performance parameters T_e , T_c , $\dot{W}$, and $\dot{Q}_c$. The input features for the backward surrogate model are the operational boundary values and performance parameters, while the output labels are the four degradation factors.

To ensure the efficient training of the MLP, all of the input features and output labels are normalized using a simple min-max normalization, as provided by the class `sklearn.preprocessing.MinMaxScaler`. Minibatch training, where the gradient is approximated in each step via a smaller set of samples, was employed to achieve faster con-

vergence [30]. In this work, 10 batches were used throughout, irrespective of the total number of training samples. Minibatch training was implemented via the PyTorch class `torch.utils.data.DataLoader`, with the data being shuffled during each epoch.

2.6. Parameter Identification

In essence, the parameter identification technique solves an optimization problem that employs the forward surrogate model, taking as input the degradation factors and calculating the plant performance parameters as output. The degradation factors serve as independent variables, and the MSE between the predicted and measured performance parameters serves as the target function.

The optimization algorithm employed here is the Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bounds (L-BFGS-B) [31]. It is a second-order quasi-Newton gradient-based method for solving large nonlinear optimization problems which are subject to simple bounds on the variables. The user must supply the gradient of the function to be minimized, but knowledge of the Hessian matrix is not required [32].

The optimization is implemented here via the `scipy.optimize.minimize` toolbox. The gradient is calculated via simple finite differencing, where each of the partial derivatives are calculated based on an upward and downward perturbation of the relevant independent variable as follows:

$$\frac{\partial J_{MSE}}{\partial f_i} = \frac{J_{MSE}(f_1, \dots, (1 + \delta h)f_j, \dots, f_n) - J_{MSE}(f_1, \dots, (1 - \delta h)f_j, \dots, f_n)}{\delta h} \quad (34)$$

where J_{MSE} is the MSE defined in Equation (33), f_j is the respective degradation factors, and δh the magnitude of the perturbation, which is set to 10^{-6} .

Several other optimization algorithms were also tested, including Nelder–Mead [33], which is a direct search method that does not require a gradient, and Differential Evolution [34], which is a stochastic population-based heuristic method. However, it was found that L-BFGS-B is robust and faster in all cases for the current application.

3. Results

In this section, we will first discuss the training data that was generated with the thermofluid model, then the process of training the forward and backward surrogate models, followed by the generation of the pseudo-measured data for condition-monitoring purposes, and finally the application of the parameter identification for the degradation detection, location, and quantification.

3.1. Training Data Generation

The trained surrogate models must be able to emulate the performance of the heat pump over the complete envelope of operation, as well as all possible combinations of the degradation factors. To ensure this, the data generation process is based on a design of experiments (DOE) using the Latin Hypercube Sampling (LHS) method [35] provided in the `pyDOE` library. The operational envelope is defined using the ranges of the input boundary values shown in Table 2. As also shown in Table 2, each of the degradation factors are allowed to vary between 0.7 and 1.1. This effectively means that the training data cover the full operating range, while the performance of any of the components may be degraded by as much as 30 percent, which can of course be adjusted down even further if required. While the 110 percent may not be physically realistic, it is conducted to provide some margin in the eventual parameter-identification process.

Table 2. Input ranges to the LHS training data generation.

	Parameter	Min	Max
Boundary values	T_{ai} [$^{\circ}\text{C}$]	15.0	35.0
	RH_i	0.10	0.90
	$\dot{m}_w$ [kg/s]	3.0	9.0
	T_{wi} [$^{\circ}\text{C}$]	15.0	35.0
Degradation factors	f_m	0.7	1.1
	f_{η}	0.7	1.1
	f_{ev}	0.7	1.1
	f_{cd}	0.7	1.1

In most situations where purely data-driven surrogate models are developed, the size of the available training data set would be a given as determined by the practical situation. In this case, where the training data are generated via a thermofluid model, the data size is only really limited by the time allocated to the data generation. Therefore, an exploratory training exercise was conducted to decide how many data points will be needed. Different sizes of data sets were used to train an MLP with three layers, each having 16 neurons. LHS was conducted for each case, with the data size varying between 100 and 4000 data points. The MLP was then trained on the different data sets for 10,000 epochs with the data split 70:30:10 between the training, validation, and testing sets. Figure 12 shows the resultant MSE for the training, validation, and testing samples for the different data set sizes. Based on this, it was decided to use a total data set size of 4000 throughout since it provides the lowest MSE values for the training, validation, and test losses.

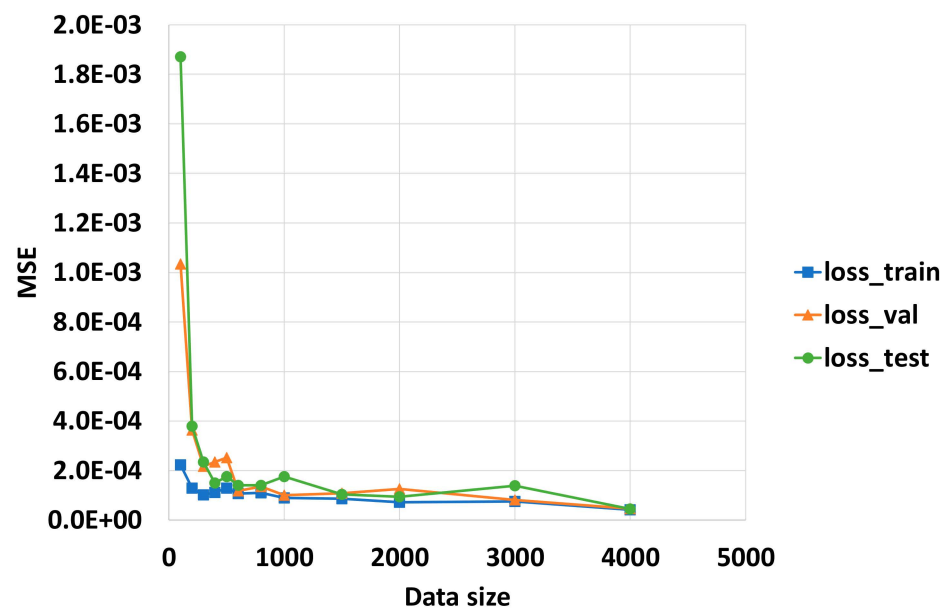
**Figure 12.** MSE for the training, validation, and testing samples after training for 10,000 epochs with different data set sizes.

Figure 13 shows histograms of the normalized inputs and outputs of the 4000 data points that were generated with the thermofluid model via the LHS. Note that the inputs are distributed almost uniformly in 10 batches, with approximately 400 data points in each. This shows the efficacy of the LHS in distributing the inputs uniformly over the different input ranges. The output data sets approximate normal distributions, as would be expected from data generated via a fundamental physics-based thermofluid model. The total run time of the thermofluid model to generate these 4000 data points was approximately 9 h.

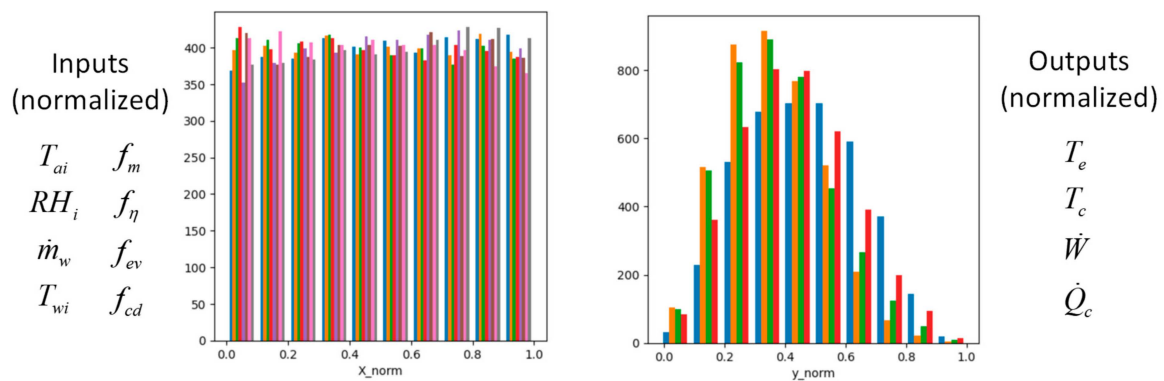


Figure 13. Histograms of the normalized input feature and output label values used in the surrogate model training.

3.2. Surrogate Model Training

A simple sequential hyperparameter search was conducted for both the forward and backward surrogate model training to determine the appropriate learning rate and neural network topology. An MLP with three layers, each having 16 neurons, was first used to determine the appropriate learning rate, whereafter the network topology was varied while the learning rate was fixed, according to the ranges shown in Table 3.

Table 3. Hyperparameter search space for MLP surrogate model training.

Parameter	Hyperparameter Search Space
Learning rate	1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2
Number of hidden layers	2, 3, 4
Neurons per hidden layer	16, 32, 48, 64

3.2.1. Forward Surrogate Model Training

Figure 14 shows the MSE for the training, validation, and testing samples after 10,000 epochs in the forward surrogate model training using different learning rates. Based on this, the learning rate for the forward surrogate model training was fixed on 1e-4 since it provides quick convergence, combined with losses comparable to the smaller learning rates.

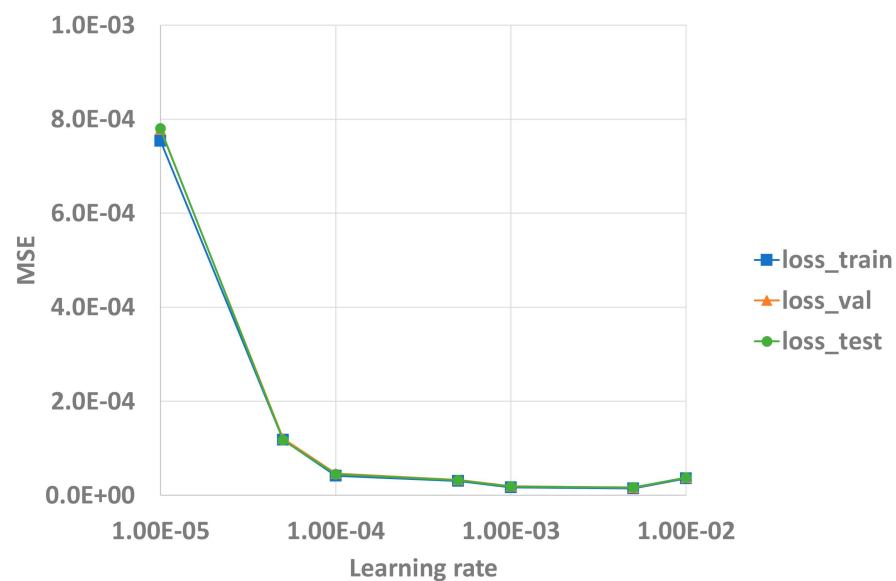


Figure 14. MSE for the training, validation, and testing samples after 10,000 epochs in the forward surrogate model training using different learning rates.

Figure 15 shows the MSE for the training, validation, and testing samples after 10,000 epochs in the forward surrogate model training using different network topologies. It shows that comparable losses are obtained for the networks with two hidden layers and sixty-four neurons each and those with four hidden layers and sixty-four neurons each. The layout with two hidden layers and sixty-four neurons each was selected for the forward surrogate model.

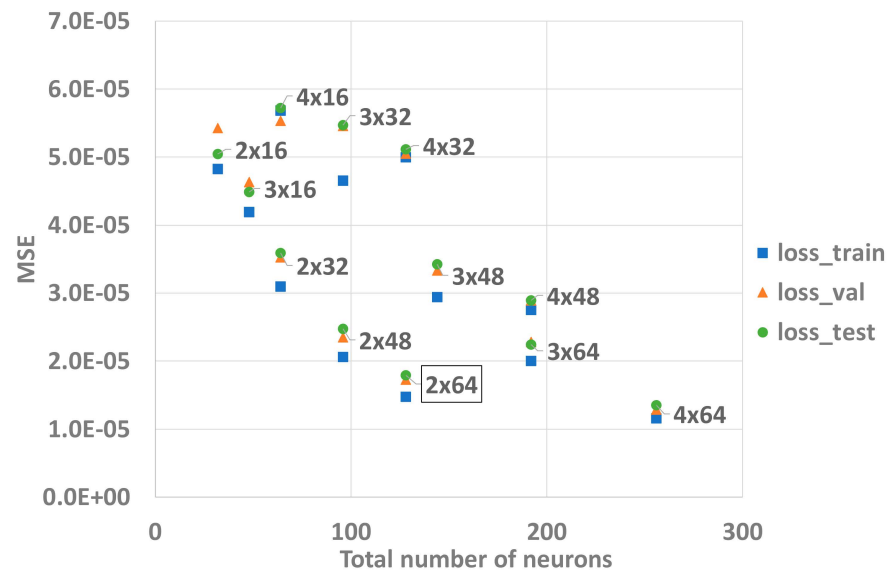


Figure 15. MSE for the training, validation, and testing samples after 10,000 epochs in the forward surrogate model training using different network topologies.

Following the hyperparameter search, the selected MLP was trained for 50,000 epochs to arrive at the final forward surrogate model used to demonstrate the condition-monitoring methodology. Figure 16 shows the history of the training and validation losses versus the number of epochs, while Table 4 shows the average and maximum percentage errors for the heat pump performance parameters predicted with the final trained forward surrogate model. It shows that good accuracy is obtained with the forward surrogate model.

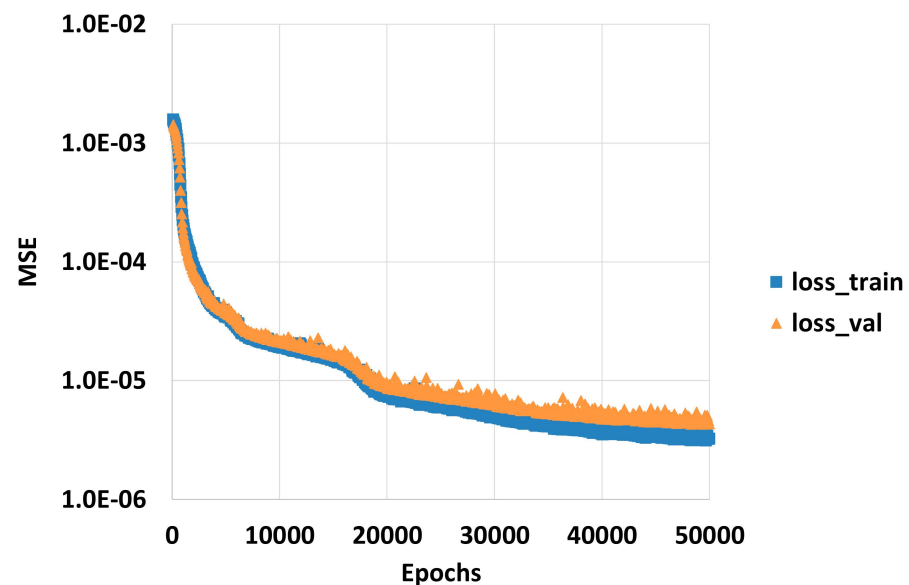


Figure 16. Training history for the forward 2×64 MLP using 4000 data points showing the training (blue squares) and validation (orange triangles) MSE versus the number of training epochs.

Table 4. Average and maximum percentage errors for the heat pump performance parameters predicted with the trained 2×64 forward surrogate model.

	T_e	T_c	$\dot{W}$	$\dot{Q}_c$
Average error	0.012%	0.016%	0.165%	0.103%
Maximum error	0.082%	0.130%	1.268%	1.011%

3.2.2. Backward Surrogate Model Training

Figure 17 shows the MSE for the training, validation, and testing samples after 10,000 epochs in the backward surrogate model training using different learning rates. Based on this, the learning rate for the backward surrogate model training was fixed at $5e-4$.

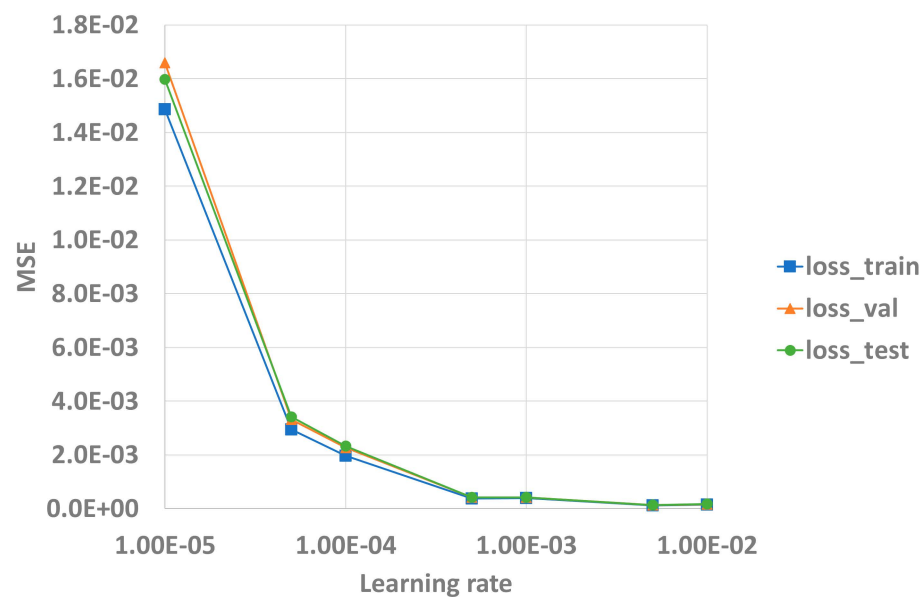
**Figure 17.** MSE for the training, validation, and testing samples after 10,000 epochs in the backward surrogate model training using different learning rates.

Figure 18 shows the MSE for the training, validation, and testing samples after 10,000 epochs in the backward surrogate model training using different network topologies. Based on this, the layout with three hidden layers and forty-eight neurons each was selected for the backward surrogate model.

Following the hyperparameter search, the selected MLP was trained for 50,000 epochs to arrive at the final backward surrogate model. Figure 19 shows the history of the training and validation losses versus the number of epochs, while Table 5 shows the average and maximum percentage errors for the degradation factors predicted with the final trained backward surrogate model.

Table 5. Average and maximum percentage errors for the degradation factors predicted with the trained 3×48 backward surrogate model.

	f_m	f_η	f_{ev}	f_{cd}
Average error	0.062%	0.102%	0.324%	0.371%
Maximum error	0.412%	0.979%	2.175%	3.096%

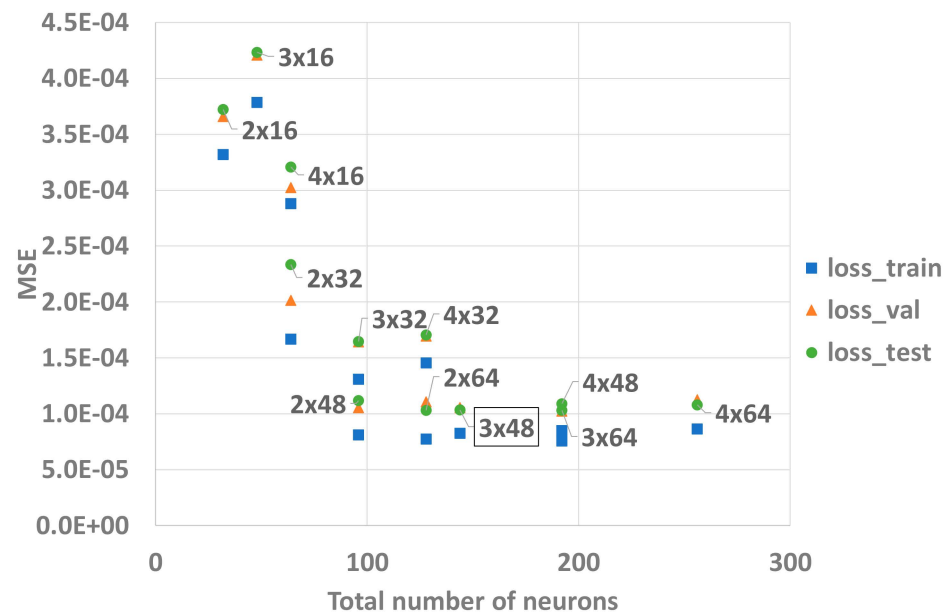


Figure 18. MSE for the training, validation, and testing samples after 10,000 epochs in the backward surrogate model training using different network topologies.

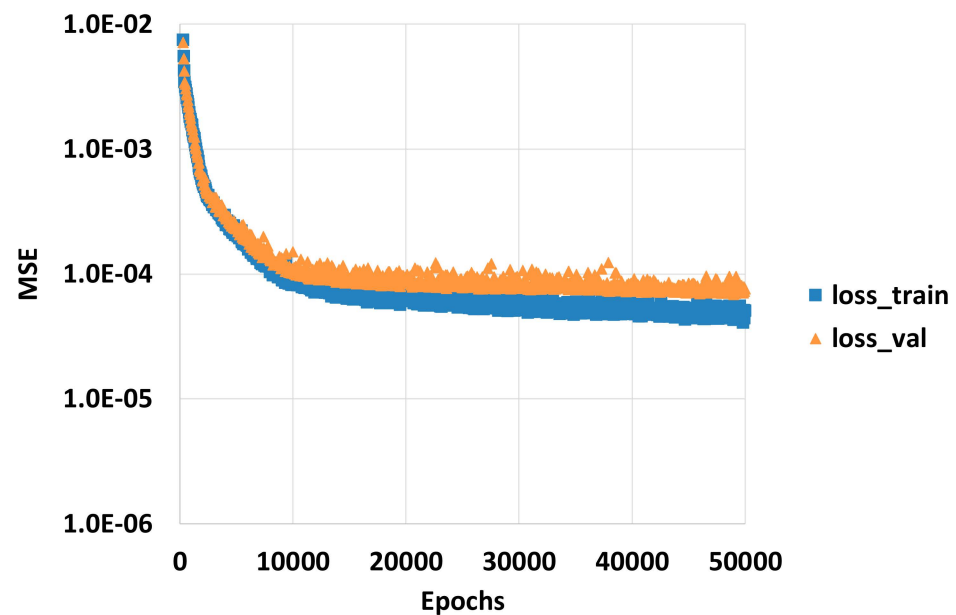


Figure 19. Training history for the 3×48 backward MLP using 4000 data points showing the training (blue) and validation (orange) MSE versus the number of training epochs.

Although the accuracy obtained via the backward surrogate model is sufficient, a comparison between Figures 16 and 19 shows that the final value of the training MSE for the backward surrogate model ($\approx 5.0 \times 10^{-5}$) is an order of magnitude greater than that of the forward model ($\approx 3.2 \times 10^{-6}$). This is also reflected in the errors reported in Table 5, which are typically three or more times than that of Table 4. It therefore proved much more challenging to train the backward model when compared to the forward model. The reason for this difference is not clear to the authors, but it is suspected that there may be a degree of multi-modality involved in the backward model. In other words, even though each unique set of input data to the thermofluid model (which is mirrored by the forward surrogate model) will only produce one unique set of output data, it may be possible that another set of input data could produce the same set of output data. Therefore, in the backward direction, a given set of input data may result in more than one possible set of output data.

In that case, using a Mixture Density Network (MDN) [30,36] might be more appropriate than a standard MLP. However, this will not be explored further in the current work.

3.3. Pseudo-Measured Plant Data Generation

The condition-monitoring methodology is based on comparing the predicted values with that of the online measurements taken on the real plant during operation. To demonstrate this, the thermofluid model was used to generate pseudo-measured plant data. A single state of degradation will be used for demonstration purposes. However, it will include simultaneous degradation via all four of the selected degradation factors. The selected values are $f_m = 0.75$, $f_\eta = 0.85$, $f_{ev} = 0.95$, and $f_{cd} = 0.80$.

Table 6 shows the operating conditions predicted via the thermofluid model for the healthy and degraded heat pump at the nominal operating point. Note that the performance of the compressor has changed significantly, with the refrigerant mass flow rate decreasing by 18.4% and the isentropic efficiency η_s being reduced by 21.9%. This does not correspond directly with the percentages of degradation imposed by f_m and f_η , which are 25% and 15%, respectively. The cooling capacity $\dot{Q}_e$ has decreased by 18.1% compared to the 5.0% imposed by f_{ev} , while the heating capacity $\dot{Q}_c$ decreased by 15.9% compared to the 20.0% imposed by f_{cd} . This shows that the integrated nature of the system does not allow a simple diagnosis to quantify the degradation in specific components.

Table 6. Nominal operating conditions of the healthy (orig) and degraded (degr) heat pump predicted via the thermofluid model.

Compressor			Condenser			Evaporator			Cycle		
	orig	degr		orig	degr		orig	degr		orig	degr
$\dot{m}_r$ [kg/s]	1.889	1.542	$\dot{m}_w$ [kg/s]	6.090	6.090	$\dot{m}_a$ [kg/s]	11.4	11.4	ΔT_{sc} [°C]	5.4	5.4
p_e [kPa]	696.2	756.9	p_w [kPa]	386.0	386.0	p_a [kPa]	86.0	86.0	ΔT_{sh} [°C]	10.0	10.0
p_c [kPa]	1653.2	1678.0	T_{wi} [°C]	20.0	20.0	T_{ai} [°C]	30.0	30.0	T_e [°C]	12.4	15.1
$\dot{W}$ [kW]	68.5	65.8	T_{we} [°C]	35.9	33.4	T_{ae} [°C]	22.7	23.9	T_c [°C]	42.7	43.3
η_s	0.607	0.474	$\dot{Q}_c$ [kW]	403.8	339.7	RH_i	0.90	0.90	$COP_{cooling}$	4.9	4.2
						RH_e	0.99	0.99			
						$\dot{Q}_e$ [kW]	335.3	274.4			

An LHS design of experiments is again employed to generate the input values so that the pseudo-measured data will uniformly cover a wide envelope of operating conditions, while each of the four degradation factors are fixed at the selected unique value. The ranges employed for the data generation are provided in Table 7. A data set of 1200 data points was generated, from which a suitable number of data points can be selected randomly as required.

Table 7. Input ranges to the LHS pseudo plant data generation.

	Parameter	Min	Max
Boundary values	T_{ai} [°C]	20.0	30.0
	RH_i	0.20	0.80
	$\dot{m}_w$ [kg/s]	4.0	8.0
	T_{wi} [°C]	20.0	30.0

3.4. Degradation Detection, Location and Quantification

Degradation detection via the forward surrogate model is conducted by solving the optimization problem, with the degradation factors as the independent variables and the MSE between the predicted and measured performance parameters for all data sets as the target function. It identifies specific areas of degradation since each degradation factor is assigned to a specific physical phenomenon. It can also quantify the degradation via the magnitude of the identified degradation factor and its deviation from the healthy value of

one. The parameter-identification process provides as output a single value for each of the degradation factors.

The backward surrogate model takes as input the measured operational boundary values as well as the measured performance parameters, and directly predicts the values of the degradation factors for each data set. The mean value of each of the degradation factors is then taken as the representative value for quantifying degradation.

To estimate the sensitivity of the proposed methodology to the number of measured data points as well as measurement uncertainty, both the forward and backward methods were applied to 15 different cases. These cases cover the use of 10, 100, and 1000 data points, each combined with measurements affected by noise sampled from a normal distribution and standard deviations of 0%, 1%, 5%, 10%, and 20%, respectively. Each data set was randomly drawn from the 1200 points presented in the previous section. The artificial noise was applied to the plant performance parameters as well as the operational boundary values, since all of these will be measured on the plant and therefore be subjected to measurement uncertainty.

Figure 20 shows a snapshot of the case employing 1000 measured data points with 10% artificial noise superimposed on all the data points. The graphs show the original data without noise on the horizontal axes versus the data with noise on the vertical axes. The graphs in the top row are for the measured operational boundary values T_{ai} , RH_i , $\dot{m}_w$, and T_{wi} , and the graphs in the bottom row are for the measured plant performance parameters T_e , T_c , $\dot{W}$, and $\dot{Q}_c$.

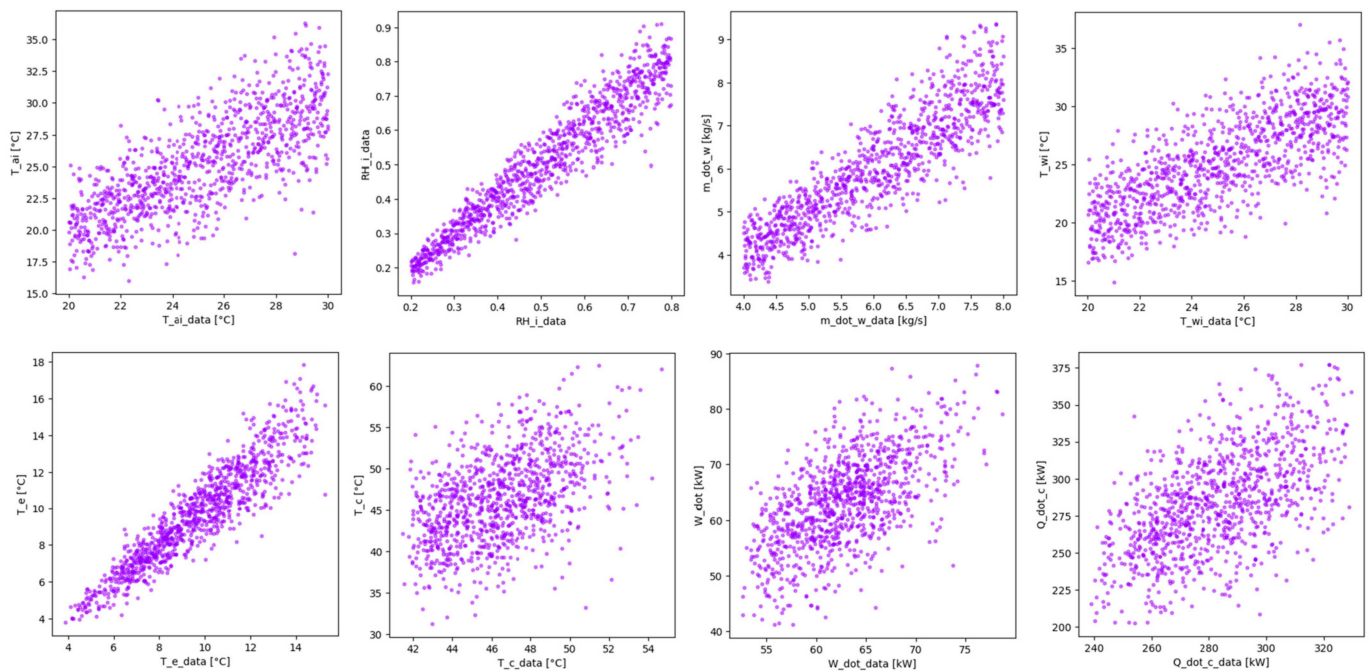


Figure 20. Snapshot of the 1000 pseudo-measured data points with artificial noise sampled from a random distribution with a standard deviation of 10% superimposed on all the data points.

3.4.1. Forward Approach

Figure 21 shows the results of the predicted versus actual degradation factors, using the forward surrogate model combined with parameter identification for the different sizes of data sets and for different levels of noise. The actual values are on the horizontal axis and the predicted values on the vertical axis. Points on the solid diagonal line means perfect prediction accuracy, while the dashed lines represent deviations of $\pm 5\%$ and $\pm 10\%$, respectively. As expected, the prediction accuracy deteriorates with less data points and more noise. The runtime for performing the prediction with 1000 data points is approximately 0.26 s.

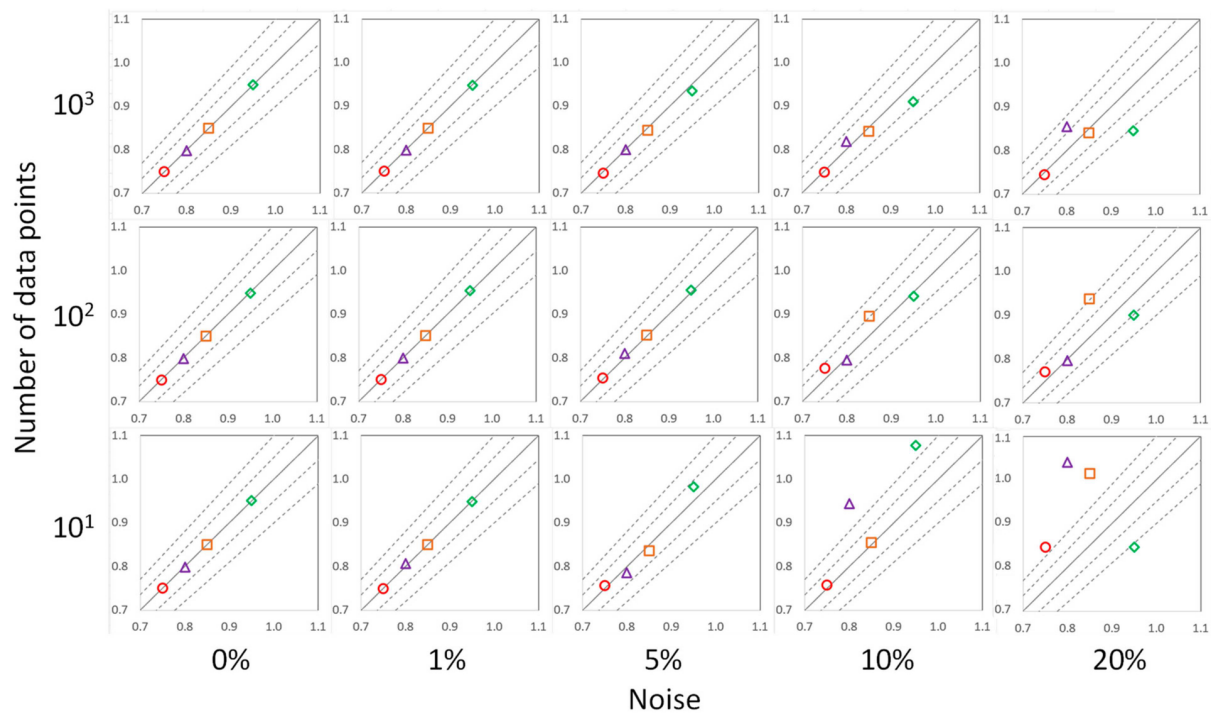


Figure 21. Results of predicted versus actual degradation factors using the forward surrogate model combined with parameter identification, with f_m —red circle, f_η —brown square, f_{ev} —green diamond, f_{cd} —purple triangle.

Figure 22 provides a summary of the prediction accuracy in the form of the average absolute percentage error between the predicted and actual values of the degradation factors for each case. For both 1000 and 100 data points, very good accuracy with an average prediction error below five percent is achieved, even with as much as 20% noise. Somewhat surprisingly, with noise up to 5%, the average prediction error is also below 5% on average with as few as 10 data points.

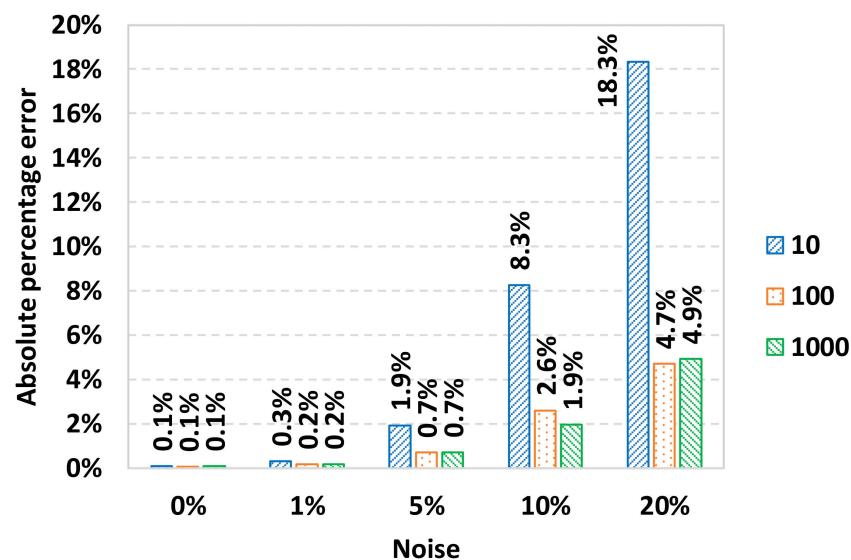


Figure 22. Average absolute percentage error between the predicted and actual values of the degradation factors for each case using the forward surrogate model combined with parameter identification.

3.4.2. Backward Approach

Figure 23 shows the results of the predicted versus actual degradation factors, using the direct backward surrogate model combined for the different sizes of data sets and for different levels of noise. Like the forward approach, the prediction accuracy deteriorates with less data points and more noise. The runtime for performing the prediction with 1000 data points is approximately 0.002 s. This is more than 100 times faster than the forward approach.

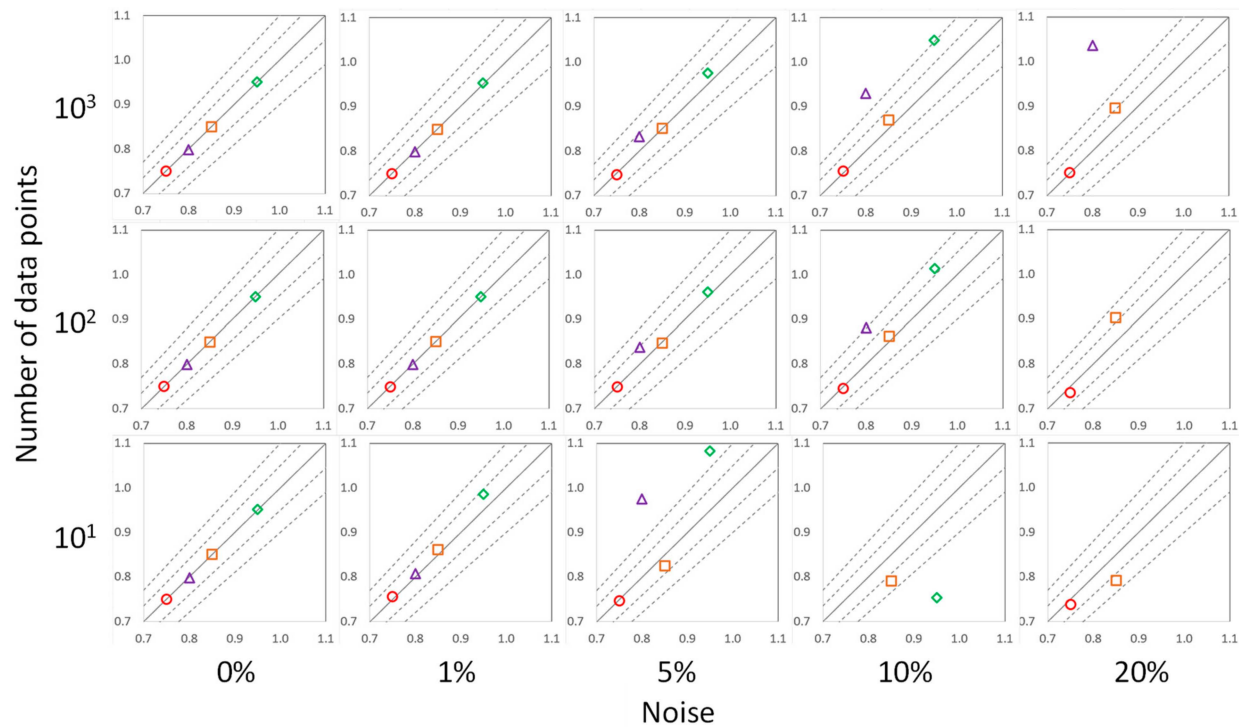


Figure 23. Results of predicted versus actual degradation factors using the backward surrogate model, with f_m —red circle, f_η —brown square, f_{ev} —green diamond, f_{cd} —purple triangle.

Figure 24 provides a summary of the prediction accuracy for the backward method. For the larger data sets with 10% noise, the prediction error approaches or exceeds five percent. For only 10 data points, the noise should be less than 5% to still achieve good accuracy. A comparison between Figure 24 with Figure 22 shows that, in general, the backward approach is less accurate and much more sensitive to noise than the forward approach.

3.4.3. Impact of the Number of Measured Performance Parameters

It was shown above that the backward approach is more than 100 times faster than the forward approach, even though the forward approach is still “fast enough”, with a typical total calculation time of less than 0.3 s. The forward approach was also shown to be more accurate and less sensitive to measurement uncertainties. The forward approach has another advantage, namely that the trained forward surrogate model and parameter identification can still be employed, even if some of the plant performance parameter measurements are not available. In other words, the forward surrogate model does not have to be retrained since the available input features are still the same. In this case, the parameter identification target function will simply exclude the output label for which the data are not available. The reduced number of output labels may of course have an impact on the prediction accuracy and the sensitivity to measurement uncertainties.

The same cannot be said for the backward approach. If one or more of the performance parameters are not available, the backward surrogate model will have to be retrained with the reduced number of input features.

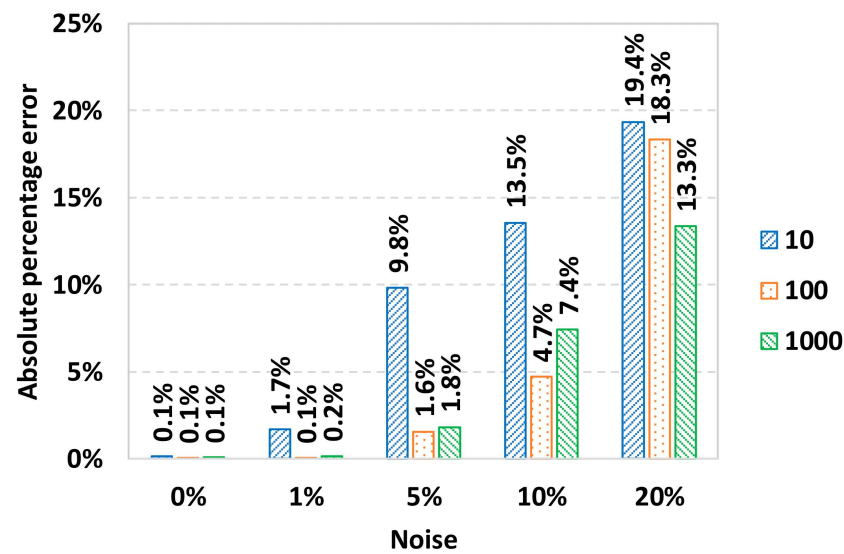


Figure 24. Average absolute percentage error between the predicted and actual values of the degradation factors for each case directly using the backward surrogate model.

Two cases were explored to demonstrate the impact of having less performance parameters available in the forward approach. In the first case, we will only employ the pseudo-measured values of T_e , T_c , and $\dot{W}$, and, for the second case, only those of T_e , $\dot{W}$, and $\dot{Q}_c$. Figure 25 shows a summary of the results for the first case, and Figure 26 for the second case.

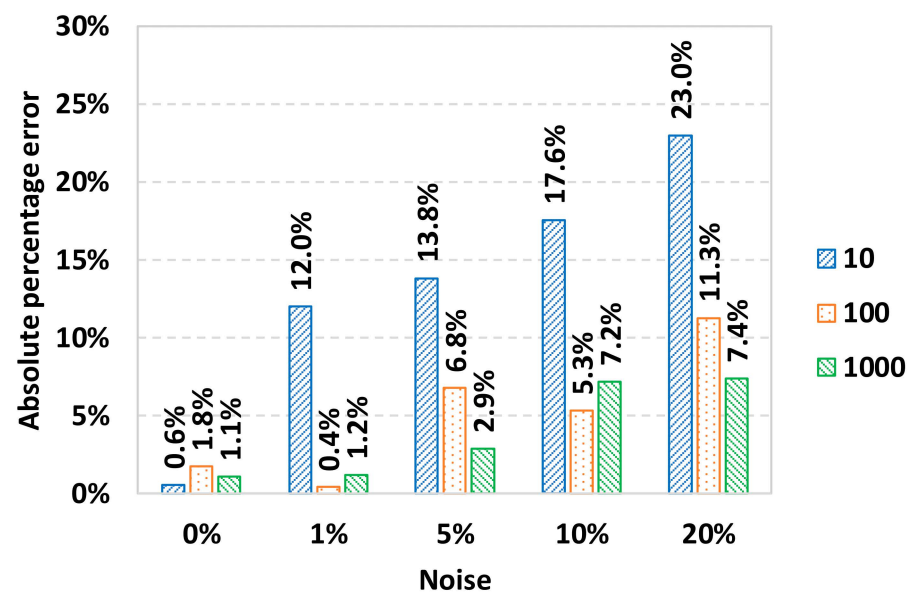


Figure 25. Average absolute percentage error between the predicted and actual values of the degradation factors for each case using the forward surrogate model combined with parameter identification, where only T_e , T_c , and $\dot{W}$ are used as output labels.

Figure 25 shows that, for the first case, the prediction error is less than 5% for the 1000 data points with 5% noise and for the 100 data points with 1% noise. For only 10 data points, the addition of any noise results in high prediction errors. For the second case, as shown in Figure 26, the prediction errors remain low for all the data sizes with up to 10% noise. These results show that, if the measurement uncertainty is low, the forward approach can still be employed with good accuracy in some cases, despite the loss of one measured performance parameter. However, the differences in the results between the

two cases show that it is not only the number of measured performance parameters that is important, but also the specific combination of parameters.

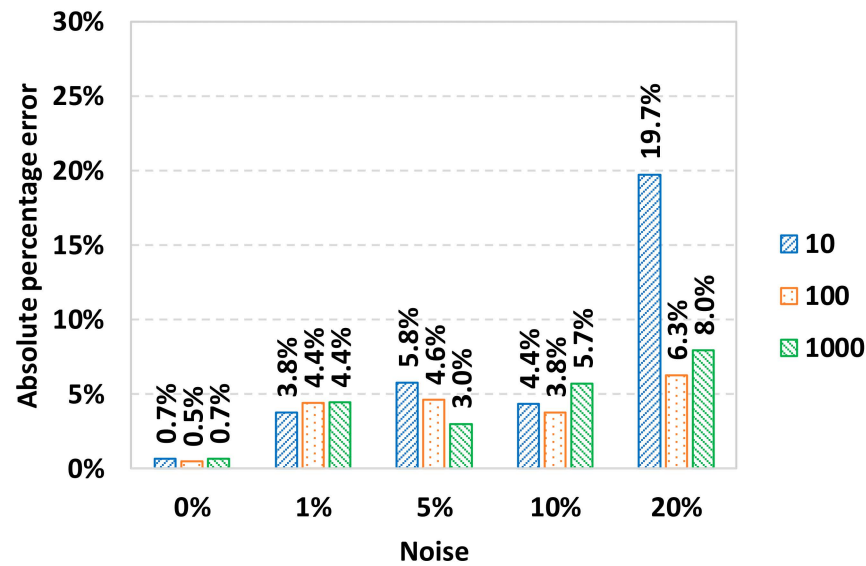


Figure 26. Average absolute percentage error between the predicted and actual values of the degradation factors for each case using the forward surrogate model combined with parameter identification, where only T_e , $\dot{W}$, and $\dot{Q}_c$ are used as output labels.

4. Conclusions

The results show that it is possible to train a forward MLP deep neural network surrogate model based on synthetic measured data, generated via a physics-based thermofluid model to predict the performance of an ACU heat pump system over a wide range of operating conditions and combinations of degradation factors. Combining this surrogate model with a suitable parameter identification technique that minimizes the residuals between the model outputs and plant measurements allows for the accurate detection, location, and quantification of degradation occurring in the different system components. For the forward approach using all four measured performance parameters, very good prediction accuracy is achieved with 100 or more measured data points, even with as much as 20% noise. Very good accuracy is also achieved with as few as 10 measured data points with noise up to 5%. The trained surrogate model can still be applied, even if some of the measured performance parameters are no longer available, for instance, due to sensor failure. However, accuracy is reduced with less measured data points, more noise, and less measured parameters.

It is also possible to train a backward MLP deep neural network surrogate model using the same training data to directly predict the degradation factors using measured plant performance parameters as input. However, the backward surrogate model proved much more challenging to train when compared to the forward model, which may indicate a degree of multi-modality involved in the backward model. The backward approach is more than 100 times faster than the forward approach, but is less accurate and much more sensitive to noise than the forward approach. Even so, the forward approach is fast enough so that the calculation time does not impede its application in practice.

Essential for the success of the methodology is a physics-based thermofluid model of the heat pump cycle that represents the characteristics of the individual components, with sufficient detail to capture the effects of degradation on its performance, as well as the integration amongst the different components. Furthermore, it must be properly calibrated with the aid of healthy plant measurements to ensure that it can accurately emulate the performance of the plant before it is used to generate the training data for the respective surrogate models.

Author Contributions: Conceptualization, P.R. and R.L.; methodology, P.R. and R.L.; software, P.R.; formal analysis, P.R.; writing—original draft preparation, P.R.; writing—review and editing, R.L.; visualization, P.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article can be made available by the authors on request.

Acknowledgments: The authors would like to acknowledge M-Tech Industrial (Pty) Ltd. in South Africa for providing technical information regarding the case study heat pump cycle.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Surucu, O.; Gadsden, S.A.; Yawney, J. Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent Advances. *Expert Syst. Appl.* **2023**, *221*, 119738. [\[CrossRef\]](#)
2. Hu, Z.; Chen, B.; Chen, W.; Tan, D.; Shen, D. Review of model-based and data-driven approaches for leak detection and location in water distribution systems. *Water Supply* **2021**, *21*, 3282–3306. [\[CrossRef\]](#)
3. Miyata, S.; Lim, J.; Akashi, Y.; Kuwahara, Y.; Tanaka, K. Fault detection and diagnosis for heat source system using convolutional neural network with imaged faulty behavior data. *Sci. Technol. Built Environ.* **2019**, *26*, 52–60. [\[CrossRef\]](#)
4. Zhao, Y.; Li, T.; Zhang, X.; Zhang, C. Artificial intelligence-based fault detection and diagnosis methods for building energy systems: Advantages, challenges and the future. *Renew. Sustain. Energy Rev.* **2019**, *109*, 85–101. [\[CrossRef\]](#)
5. Hou, Z.; Lee, C.K.M.; Lv, Y.; Keung, K.L. Fault detection and diagnosis of air brake system: A systematic review. *J. Manuf. Syst.* **2023**, *71*, 34–58. [\[CrossRef\]](#)
6. Guo, S.; Wang, J.; Wei, J.; Zachariades, P. A new model-based approach for power plant Tube-ball mill condition monitoring and fault detection. *Energy Convers. Manag.* **2014**, *80*, 10–19. [\[CrossRef\]](#)
7. Gao, Y.; Miyata, S.; Akashi, Y. Automated fault detection and diagnosis of chiller water plants based on convolutional neural network and knowledge distillation. *Build. Environ.* **2023**, *245*, 110885. [\[CrossRef\]](#)
8. Gao, L.; Li, D.; Liang, N. Genetic algorithm-aided ensemble model for sensor fault detection and diagnosis of air-cooled chiller system. *Build. Environ.* **2023**, *233*, 110089. [\[CrossRef\]](#)
9. Zhang, L.; Cheng, Y.; Zhang, J.; Chen, H.; Cheng, H.; Gou, W. Refrigerant charge fault diagnosis strategy for VRF systems based on stacking ensemble learning. *Build. Environ.* **2023**, *234*, 110209. [\[CrossRef\]](#)
10. Du, Z.; Fan, B.; Jin, X.; Chi, J. Fault detection and diagnosis for buildings and HVAC systems using combined neural networks and subtractive clustering analysis. *Build. Environ.* **2014**, *73*, 1–11. [\[CrossRef\]](#)
11. Underwood, C.P.; Royapoor, M.; Sturm, B. Parametric modelling of domestic air-source heat pumps. *Energy Build.* **2017**, *139*, 578–589. [\[CrossRef\]](#)
12. Catano, J.; Zhang, T.; Wen, J.T.; Jensen, M.K.; Peles, Y. Vapor compression refrigeration cycle for electronics cooling—Part I: Dynamic modeling and experimental validation. *Int. J. Heat Mass Transf.* **2013**, *66*, 911–921. [\[CrossRef\]](#)
13. Kocyigit, N.; Bulgurcu, H.; Lin, C.-X. Fault diagnosis of a vapor compression refrigeration system with hermetic reciprocating compressor based on p-h diagram. *Int. J. Refrig.* **2014**, *45*, 44–54. [\[CrossRef\]](#)
14. Lee, K.-P.; Wu, B.-H.; Peng, S.-L. Deep-learning-based fault detection and diagnosis of air-handling units. *Build. Environ.* **2019**, *157*, 24–33. [\[CrossRef\]](#)
15. Miyata, S.; Kuwahara, Y.; Tsunemoto, S.; Tanaka, K.; Akashi, Y. Improving training efficiency for scalable automated fault detection and diagnosis in chilled water plants by transfer learning. *Energy Build.* **2023**, *285*, 112877. [\[CrossRef\]](#)
16. Kim, M.; Yoon, S.H.; Payne, W.V.; Domanski, P.A. Development of the reference model for a residential heat pump system for cooling mode fault detection and diagnosis. *J. Mech. Sci. Technol.* **2010**, *24*, 1481–1489. [\[CrossRef\]](#)
17. Aguilera, J.J.; Meesenburg, W.; Ommen, T.; Markussen, W.B.; Poulsen, J.L.; Zühlsdorf, B.; Elmegaard, B. A review of common faults in large-scale heat pumps. *Renew. Sustain. Energy Rev.* **2022**, *168*, 112826. [\[CrossRef\]](#)
18. Rogers, A.P.; Guo, F.; Rasmussen, B.P. A review of fault detection and diagnosis methods for residential air conditioning systems. *Build. Environ.* **2019**, *161*, 106236. [\[CrossRef\]](#)
19. Li, H.; Braun, J.E. Decoupling features and virtual sensors for diagnosis of faults in vapor compression air conditioners. *Int. J. Refrig.* **2007**, *30*, 546–564. [\[CrossRef\]](#)
20. Piacentino, A.; Talamo, M. Critical analysis of conventional thermoeconomic approaches to the diagnosis of multiple faults in air conditioning units: Capabilities, drawbacks and improvement directions. A case study for an air-cooled system with 120 kW capacity. *Int. J. Refrig.* **2013**, *36*, 24–44. [\[CrossRef\]](#)
21. Wang, F.; Guo, J.; Jiang, Y.; Sun, C. Parameter identification framework of thermal network model for ventilated heating floor. *Energy Build.* **2024**, *311*, 114138. [\[CrossRef\]](#)
22. Shi, D.; Xu, Y.; Demartino, C.; Xiao, Y.; Spencer, B.F. Cyclic behavior of laminated bio-based connections with slotted-in steel plates: Genetic algorithm, deterministic neural network-based model parameter identification, and uncertainty quantification. *Eng. Struct.* **2024**, *310*, 118114. [\[CrossRef\]](#)

23. Yin, X.; Wen, K.; Huang, W.; Luo, Y.; Ding, Y.; Gong, J.; Gao, J.; Hong, B. A high-accuracy online transient simulation framework of natural gas pipeline network by integrating physics-based and data-driven methods. *Appl. Energy* **2023**, *333*, 120615. [[CrossRef](#)]
24. Ghadbane, H.E.; Rezk, H.; Ferahtia, S.; Barkat, S.; Al-Dhaifallah, M. Optimal parameter identification strategy applied to lithium-ion battery model for electric vehicles using drive cycle data. *Energy Rep.* **2024**, *11*, 2049–2058. [[CrossRef](#)]
25. Yin, Y.; He, Q.; Hu, F.; Huang, W. Merging experiment data and simulation data for parameter identification of shaft seal. *Measurement* **2024**, *236*, 114863. [[CrossRef](#)]
26. Jahnig, D.T.; Reindl, D.T.; Klein, S.A. Semi-empirical method for representing domestic refrigerator/freezer compressor calorimeter test data. In Proceedings of the ASHRAE Transactions, Minneapolis, MN, USA, 25–28 June 2000.
27. Rousseau, P.; Laubscher, R.; Rawlins, B.T. Heat Transfer Analysis Using Thermofluid Network Models for Industrial Biomass and Utility Scale Coal-Fired Boilers. *Energies* **2023**, *16*, 1741. [[CrossRef](#)]
28. Rousseau, P.G.; Van Eldik, M.; Greyvenstein, G.P. Detailed simulation of fluted tube water heating condensers. *Int. J. Refrig.* **2003**, *26*, 232–239. [[CrossRef](#)]
29. Jones, W.P. *Air Conditioning Engineering*, 5th ed.; Elsevier Butterworth-Heinemann: Oxford, UK, 2005.
30. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*, 1st ed.; MIT Press: Chennai, India, 2017.
31. Byrd, R.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.* **1995**, *16*, 1190–1208. [[CrossRef](#)]
32. Zhu, H.; Byrd, R.; Nocedal, J. Algorithm 778 L-BFGS-B Fortran subroutines for large scale bound constrained optimizations. *ACM Trans. Math. Softw.* **1997**, *23*, 550–560. [[CrossRef](#)]
33. Gao, F.; Han, L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.* **2010**, *51*, 259–277. [[CrossRef](#)]
34. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
35. Stein, M. Large sample properties of simulations using latin hypercube sampling. *Technometrics* **1987**, *29*, 143–151. [[CrossRef](#)]
36. Rawlins, B.T.; Laubscher, R.; Rousseau, P. An integrated data-driven surrogate model and thermofluid network-based model of a 620 MWe utility-scale boiler. *Proc. Inst. Mech. Eng. Part A J. Power Energy* **2022**, *237*, 1061–1079. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.