Institut für
Strahlantriebe und
Turbomaschinen

Bachelor Thesis

Machine Learning-Based Performance Prediction of Pulsating Heat Pipes

by Illbruck, Marvin
Student ID: 442715

This thesis was submitted to the
Institute of Jet Propulsion and Turbomachinery

Faculty of Mechanical Engineering
RWTH Aachen University

1. Examiner:	Univ.-Prof. Dr.-Ing. Peter Jeschke
2. Examiner:	Dr.-Ing. Stefan Henninger
Supervisor:	Timo Engelmann, M. Sc.

Aachen, May 5, 2026

Kurzfassung

Der anthropogene Klimawandel erhöht den Bedarf an Forschung und Entwicklung, um Treibhausgasemissionen nachhaltig zu senken. Dabei erfordert die Dekarbonisierung des Verkehrssektors, insbesondere in der Luftfahrt, innovative Ansätze zur Entwicklung und Integration alternativer Antriebskonzepte, um deren effizienten und sicheren Betrieb zu gewährleisten. Als vielversprechendes alternatives Antriebskonzept gelten Polymerelektrolytmembran-Brennstoffzellen (PEMFCs), die jedoch ein anspruchsvolles thermisches Management erfordern. In diesem Zusammenhang könnten pulsierende Wärmerohre (engl. Pulsating Heat Pipes, PHPs) einen vielversprechenden Lösungsansatz bieten. Aufgrund ihrer hohen geometrischen Gestaltungsfreiheit, kompakten Bauweise und passiver Arbeitsweise sind sie für Luft- und Raumfahrtanwendungen geeignet und können in bestehende Komponenten, wie beispielsweise die Bipolarplatten von Brennstoffzellen, integriert werden. Ihr komplexes Zweiphasenströmungsverhalten erschwert die genauere Leistungsvorhersage über numerische Berechnungsmethoden als Auslegungswerkzeug ganzer PHP-Systeme. Diese Arbeit entwickelt ein auf maschinellem Lernen (ML) basierendes Modell, um das thermische Verhalten von PHPs vorherzusagen. Der Zielparameter ist der thermische Widerstand, charakteristisch für die thermische Leistungsfähigkeit der PHP. Die Schätzungen des Modells werden abschließend mit CFD-Simulationen als alternativem Ansatz zur Leistungsprognose sowie mit experimentellen Messungen verglichen. Schlüsselparameter von PHPs wie der Wärmestrom, effektive Fläche, Füllgrad, Neigungswinkel, geometrische Merkmale (z.B. hydraulischer Durchmesser, Anzahl der Windungen) und dimensionslose Kennzahlen werden als Eingabeparameter in das ML-Modell einbezogen. Insgesamt wurden 3.473 Datenpunkte aus früheren Publikationen zu PHP-Experimenten zusammengetragen. Aufgrund seiner Genauigkeit, Interpretierbarkeit und Effizienz bei der Arbeit mit komplexen Datensätzen wird XGBoost, ein auf Gradient Boosting basierendes Modell, für die Vorhersageaufgabe ausgewählt. Die Schätzungen des Modells zeigen eine hohe Vorhersagegenauigkeit mit niedrigen Fehlermetriken wie RMSPE = 20,2% und $R^2 = 0{,}941$. Die SHAP-Analyse quantifiziert die relativen Beiträge der Eingabeparameter zur Gesamtvorhersage, wobei die Wärmestromdichte im Verdampferteil der PHP den dominierenden Einfluss auf die Modellschätzungen hat. Die darüber hinaus durchgeführten CFD-Simulationen bestätigen den hohen Rechenaufwand, der für die Modellierung von PHPs erforderlich ist. Gleichzeitig zeigt sich, dass zur Gewährleistung rechnerischer Umsetzbarkeit simulative Vereinfachungen notwendig sind, die die Übereinstimmung mit den zur Validierung herangezogenen experimentellen Messungen einschränken. Insgesamt tragen CFD-Simulationen zu einem besseren Verständnis des Verhaltens von PHPs und den zugrunde liegenden simulativen Herausforderungen bei, ihr praktischer Nutzen für die Auslegung neuer PHPs bleibt jedoch durch große Abweichungen mit den Validierungsdaten sowie dem hohen Rechenaufwand eingeschränkt. Im Gegensatz dazu bietet das ML-Modell die erforderliche Geschwindigkeit und Zuverlässigkeit für eine großflächige Designoptimierung und stellt damit einen Fortschritt in der Auslegung und Analyse von PHPs für die Kühlung von Luft- und Raumfahrtsystemen dar.

Abstract

Anthropogenic climate change increases the need for research and development aimed at reducing greenhouse gas emissions. In the transportation sector, and particularly in aviation, decarbonization requires innovative approaches to ensure the safe and efficient operation of alternative propulsion systems. Proton exchange membrane fuel cells (PEMFCs) are a promising option, but they require stringent temperature control. Pulsating heat pipes (PHPs) could represent a versatile and passive thermal management solution. Due to their high degree of design flexibility and compact construction, they are well-suited for aerospace applications and can be integrated into existing components, such as the bipolar plates of fuel cells. Their complex two-phase flow behavior, however, including small liquid films, capillary effects, and phase change processes, makes accurate performance prediction difficult when relying solely on conventional CFD simulations, which either incur high computational cost, limiting their practical applicability, or achieve lower cost at the expense of predictive accuracy. This work introduces a machine learning (ML) framework designed to predict the thermal behavior of PHPs. This data-driven approach is benchmarked against conducted CFD simulations as an alternative approach to performance prediction and experimental results, showing that while CFD simulations can provide physical insight into PHP behavior, achieving computationally feasible simulations requires compromises in accuracy, which limits their applicability for design tasks on a large scale. By utilizing the ML approach, this study provides an efficient alternative for exploring broad PHP design spaces. Key PHP parameters, such as the heat flux, effective area, fill ratio, inclination angle, geometric features (e.g., hydraulic diameter, number of turns), and dimensionless numbers, are included as input parameters in the ML model. A total of 3,473 data points were compiled through past publications on PHP experiments. XGBoost, an algorithm based on gradient boosting, is chosen for the predictive task due to its accuracy, interpretability, and efficiency when working with complex datasets. The ML model is built to predict the key indicator of the heat transfer capabilities of PHPs, which is thermal resistance. Model optimization and performance are assessed using cross-validation, Bayesian optimization, and SHAP analysis, which quantifies the relative contributions of the input features to the model predictions, revealing the heat flux as the most impactful feature. The results indicate high predictive accuracy with an RMSPE = 20.2% and $R^2 = 0.941$ and solid generalization across varying geometries and operating regimes. The results of the CFD simulations confirm the high computational effort required for simulating PHPs. Overall, while CFD contributes to an improved understanding of PHP behavior and the underlying challenges in its numerical simulation, its practical applicability remains constrained by limited accuracy and high computational requirements. In contrast, the ML model offers the necessary speed and reliability for design optimization, enabling rapid exploration of the design space and facilitating the design and integration of PHPs into complex aerospace thermal management systems.

Contents

List of Figures

List of Tables

List of Symbols

Latin symbols

Symbol	**Definition**	**Unit**
A	Area	m^2
B	Number of bins	-
Bo	Bond number	-
Br	Brinkmann number	-
c_p	Specific isobaric heat capacity	$\mathrm{J/(kg\,K)}$
D	Diameter	m
$\mathcal{D}$	Dataset	-
D_h	Hydraulic diameter	m
D_i	Inner diameter	m
e	Specific Energy	J/kg
$\hat{E}(x)$	Expectation	unit of x
f	Frequency	Hz
F	Force; Faraday constant	N; C/mol
f_k	Tree function in XGB	-
g	Gravity	$\mathrm{m/s}^2$
g_i	Gradient (first derivative of loss)	unit varies with loss function
G	Sum of gradients	unit varies with loss function
Gr	Grashof number	-
h	Enthalpy	J/kg
h_i	Hessian	unit varies with loss function
h_{lv}	Latent heat of vaporization	J/kg
H	Sum of Hessians	unit varies with loss function
i	Density of proton current	$\mathrm{A/m}^2$
I	Identity tensor; index set	-
Ja	Jakob number	-
$J_{j,q}$	Species diffusion flux	$\mathrm{kg/m^2\,s}$
K	Number of decision trees	-
Ku	Kutateladze number	-
l	Loss function; length	-; m
L	Objective function	-
L_{cd-ev}	Length between condenser and evaporator	m
L_m	Membrane thickness	m
m	Number of input features	-
$\dot{m}$	Mass transfer rate	kg/s
M	Molar mass	kg/mol
n	Molar rate	$\mathrm{mol/m^3\,s}$
N	Number of samples	-
Nu	Nusselt number	-
Oh	Ohnesorge number	-
p	Pressure	Pa
Pr	Prandtl number	-
q''	Heat flux	$\mathrm{W/m}^2$
$\dot{q}$	Volumetric heat source	$\mathrm{W/m}^3$

$\dot{Q}$	Heat transfer rate	W
R	Residual	unit of residual
R_{th}	Thermal resistance	K/W
Ra	Rayleigh number	-
Re	Reynolds number	-
R_{ohm}	Membrane resistance	$\Omega\,\mathrm{m}^2$
R^2	Coefficient of determination	-
$\dot{R}$	Interphase force per unit volume	$\mathrm{N/m}^3$
S_h	Volumetric energy source	$\mathrm{W/m}^3$
Sc	Quality score	-
t	Time	s
T	Temperature; number of leaves	K; -
u	Velocity	m/s
V	Volume	m^3
w	Weight	-
x, X	Input feature	unit of x
$y, \hat{y}$	Target value; prediction	unit of y

Greek symbols

Symbol	**Definition**	**Unit**
α	Volume fraction; heat transfer coefficient	-; $\mathrm{W/m}^2\,\mathrm{K}$
β; $\beta_{c;v}$	Thermal expansion; relaxation coefficient	1/K, 1/s
γ	Pruning parameter	-
$\hat{\gamma}_1; \hat{\gamma}_2$	Skewness; kurtosis	-
Δ	Difference (e.g., ΔT)	-
∇	Nabla operator	1/m
ϵ	Emissivity	-
$\hat{\epsilon}$	Residual	unit of residual
η	Learning rate; local overpotential	-; V
λ	Regularization; thermal conductivity	-; $\mathrm{W/(m\,K)}$
μ	Dynamic viscosity	Pa/s
ν	Kinematic viscosity	$\mathrm{m}^2\mathrm{/s}$
ρ	Density	$\mathrm{kg/m}^3$
σ	Surface tension	N/m
σ_{SB}	Stefan-Boltzmann constant	$\mathrm{W/m}^2\,\mathrm{K}^4$
$\sigma_{sd}(x)$	Std. deviation	unit of x
τ	Shear stress	Pa
φ, ϕ	Sum of tree functions; Shapley value	-
Φ	Dissipation function	$\mathrm{W/m}^3$
Ω	Regularization function	-

Indices

Symbol	**Definition**
act	Activation
amb	Ambient
avg	Average
c	Condensation
cd	Condenser
cd-ev	Condenser-evaporator
cell	Cell inside the mesh
cond	Conduction
cr	Cross-section
e	Evaporation
ev	Evaporator
h	Heat
H_2O	Water
i	Inner; index
j	Index of a species
l	Liquid
lv	Vaporization
log	Logarithm
L	Left
p	Primary phase
q	Secondary phase
R	Right
rad	Radiation
reac	Reaction
ref	Reference
sat	Saturation
SB	Stefan-Boltzmann
sd	Standard deviation
surf	Surface
v	Vapor
w	Wall

Abbreviations

Symbol	**Definition**
ANN	Artificial neural network
ARD	Automatic relevance determination
BO	Bayesian optimization
BPP	Bipolar plate
CFD	Computational fluid dynamics
CFL	Courant-Friedrichs-Lewy
CV	Cross validation
EI	Expected improvement
FS	Fuel stack

GDL	Gas diffusion layer
GP	Gaussian process
GS	Grid search
HT	High temperature
LT	Low temperature
ML	Machine learning
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MEA	Membrane electrode assembly
MSE	Mean squared error
NIST	National Institute of Standards and Technology
NN	Neural network
Q-Q	Quantile-Quantile
PBI	Polybenzimidazole
PEM	Polymer electrolyte membrane
PEMFC	Proton exchange membrane fuel cell
PHP	Pulsating heat pipe
PISO	Pressure-velocity coupling
RF	Random forest
RMSE	Root mean squared error
RMSPE	Root mean squared percentage error
RS	Random search
SD	Standard deviation
SCA	Statistical convergence assessment
SHAP	Shapley additive explanations
SEM	Standard error of the mean
TMS	Thermal management system
TPE	Tree Parzen estimator
VOF	Volume of fluid
WF	Working fluid
XGB	Extreme gradient boosting

1. Introduction and State of the Art

1.1. Motivation

Anthropogenic climate change is one of the most pressing global challenges. The Paris Agreement commits the international community to limiting global warming to below 2 °C and pursuing efforts to limit it to 1.5 °C [30, 79]. In this context, decarbonizing the transportation sector is essential. Transportation contributes significantly to global CO_2 and NO_x emissions, and increasingly strict governmental regulations are pushing industry stakeholders toward propulsion concepts with reduced emissions and climate impact. For applications requiring high energy density, such as aviation, battery systems often do not provide sufficient performance in relation to their weight. As a result, hydrogen has emerged as a promising alternative energy carrier. Owing to its high gravimetric energy density and the possibility of producing it from renewable sources, hydrogen is considered a key technology for meeting long-term climate objectives [29]. Despite its potential, several technical challenges remain. Hydrogen storage, either as compressed gas at pressures up to 700 bar, as cryogenic liquid at approximately -253 °C, or in advanced material-based storage systems, poses considerable engineering, safety, and efficiency constraints [50]. Moreover, the thermal management requirements of hydrogen-based propulsion systems, especially those incorporating proton exchange membrane fuel cells (PEMFC), are demanding. Unlike conventional combustion engines, fuel cells and batteries operate within relatively narrow temperature windows. Effective cooling is critical to avoid performance degradation, prevent local hotspots, and ensure long-term durability. In aviation, these challenges are amplified by strict weight, volume, safety, and operational requirements. Designing efficient, lightweight cooling architectures and selecting appropriate hydrogen storage concepts are crucial steps toward enabling hydrogen-powered flight. Pulsating heat pipes (PHPs), as a thermal management component, could provide an operational value for this case. Integrated, e.g., directly into bipolar plates, they allow heat to be diverted from the fuel cell stack without the parasitic power consumption required by conventional cooling systems. A lot of research has been conducted on the performance assessment of pulsating heat pipes. However, detailed numerical simulations of entire PHP systems take large amounts of computational time and resources, making it impractical for the iteration and analysis of new designs. Therefore, experiments currently remain the preferred and most practical approach for performance evaluation and design iteration. But with the increasing accessibility of artificial intelligence and machine learning (ML) in the last few decades, utilizing a machine learning algorithm trained on experimental data offers a resource-efficient way of performance prediction and analysis. The ML-based approach of this work enables the identification of the complex non-linear interactions of the used input features essential for PHP systems, enabling future designs with reduced resource requirements, facilitating the iteration and interpretation of the feasibility of PHP designs for the application in aviation and space systems. Additionally, numerical simulations are conducted to predict the thermal performance of PHPs, identify their limitations, and enhance the overall understanding of the involved physics. Thermal resistance is selected as the key indicator of PHP performance. In particular, the time-averaged thermal resistance is used as the primary performance metric, as pulsating heat pipes operate under inherently transient conditions, making this quantity the most relevant and commonly reported indicator in the literature.

1.2. Heat Transfer in PEM Fuel Cell Systems

To analyze the mechanisms of heat generation in PEMFCs, the basic structure of a typical low-temperature PEM fuel cell (LT-PEMFC), shown in Fig. 1.1, is examined. The structure consists of an anode and a cathode side. Both contain a bipolar plate that distributes the reactant gases onto a gas diffusion layer (GDL). The hydrogen and oxygen are then distributed to the catalyst layer, where the hydrogen is ionized at the anode.

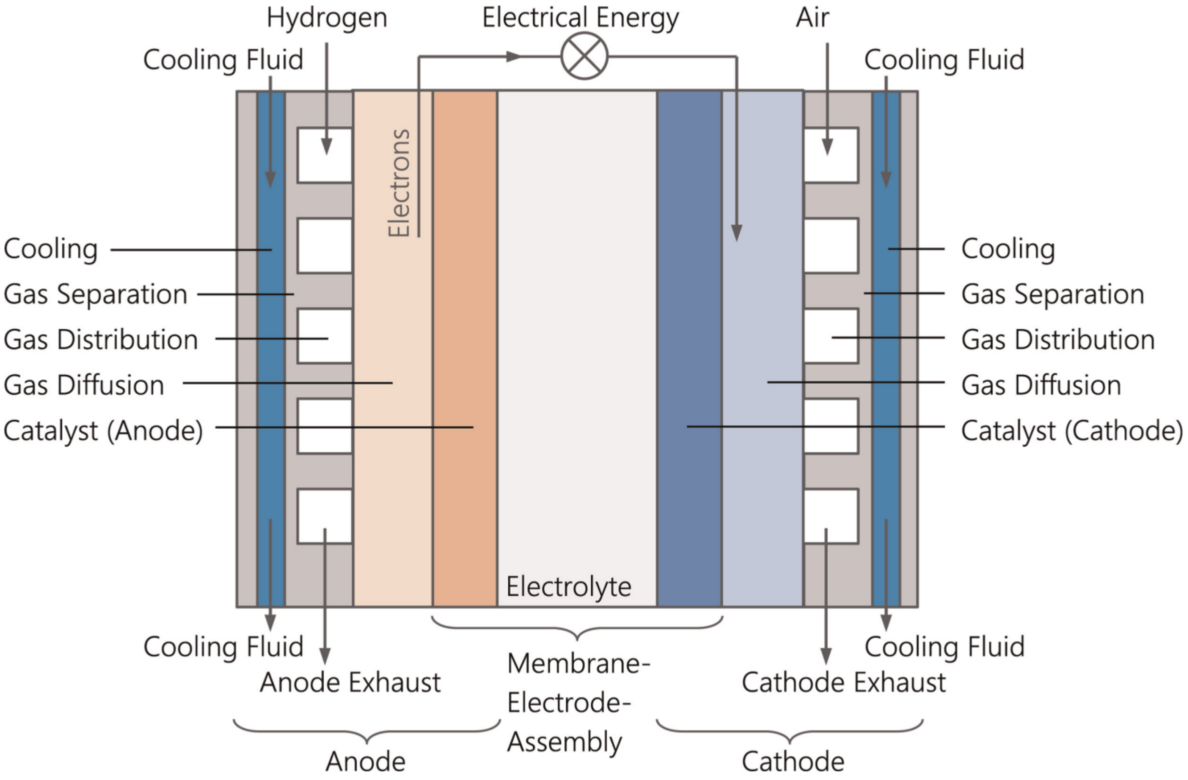

Figure 1.1: Fuel cell structure based on common LT-PEMFC designs [32]

$$H_2 \rightarrow 2H^+ + 2e^- \tag{1.1}$$

The resulting protons H^+ diffuse through the Polymer Electrolyte Membrane (PEM), while the electrons e^- must travel through an external circuit. On the cathode side, oxygen is supplied, where it accepts the electrons arriving from the anode and reacts with the protons that have crossed the membrane.

$$O_2 + 4e^- \rightarrow 2O^{2-} \tag{1.2}$$

This electrochemical reaction forms water and leaves excess air behind.

$$2H^{+} + O^{2-} \rightarrow H_2O \tag{1.3}$$

The movement of electrons from the anode to the cathode generates electrical power, which can be used to drive an electric motor. During these reactions, heat is produced. This heat can be dissipated both internally through cooling channels embedded within the PEMFC stack and externally via dedicated heat exchangers. Around 40–60% of the chemical energy is converted into heat [32]. Generally, hydrogen PEMFCs can be categorized into high temperature (HT)-PEMFCs and low temperature (LT)-PEMFCs. As the name suggests, the major difference between the two is the operating temperature. LT-PEMFCs are fairly sensitive to their temperature range and require minimized temperature gradients for efficient operation. Furthermore, humidification of the membrane is required to ensure smooth operation. Hence, complex cooling and control systems become essential, resulting in an overall increase in system complexity. Especially in aircraft systems, where the surrounding air is normally fairly dry, the drying out of the membrane can lead to increased resistance and, consequently, overheating of the stack. HT-PEMFCs involve some minor material changes in components to allow improved operation at higher temperatures of around 130–180 °C, whereas LT-PEMFCs typically operate at 60–80 °C. Some key advantages of HT-PEMFCs include the increased temperature difference to the ambient air, resulting in higher natural heat flux, which helps with cooling and maintaining the fuel cell within optimal operating temperature ranges. This allows for smaller heat exchanger components, reducing system complexity and weight. Furthermore, no humidification system is necessary, as proton conduction in the phosphoric acid network of the polybenzimidazole (PBI) membrane is water-independent [49]. PHPs need a certain amount of heat to operate, a heat input that can be achieved by the LT-PEMFCs as well as the HT-PEMFCs, indicating that PHPs are suited as a heat transfer component inside the fuel cell.

Generally, there are three major types of heat sources in PEMFCs to be considered [65]. The first one is volumetric sources within the membrane. The transport of protons suffers from resistance, known as the Joule effect, and can be calculated by:

$$\dot{q}_j = \frac{R_{ohm} i^2}{L_m} \tag{1.4}$$

with R_{ohm} being the membrane resistance, i the density of proton current, and L_m the membrane thickness. This effect is equally distributed over the membrane. Secondly, heat is released at the electrodes due to the half-reactions of hydrogen oxidation at the anode and oxygen reduction at the cathode. This includes the reversible reaction heat, which scales with current density as:

$$q''^{,i}_{\text{reac}} = iQ^{i}_{\text{reac}}/2F \tag{1.5}$$

and the activation heat due to overpotentials expressed as:

$$q''^{,i}_{\text{act}} = \eta_i i \tag{1.6}$$

where η_i is the local overpotential. Third, water produced at the cathode may evaporate or condense within the gas diffusion layer, generating additional heat or absorbing heat from the surroundings. This contribution is calculated from the latent heat of water as:

$$\dot{q}_{c/e} = n_{\text{v}\rightarrow\text{l}} M_{H_2O} h_{lv} \tag{1.7}$$

with h_{lv} being the latent heat, $n_{\mathrm{v}\to\mathrm{l}}$ the molar rate of condensed or evaporated water, and M_{H_2O} the molar mass of water.

The generated heat is transferred via three primary mechanisms: conduction, convection, and radiation. These processes govern the spatial and temporal distribution of temperature within the stack, directly influencing performance, durability, and safety [85]. The dominant mode of heat transfer varies across the components described earlier. The conductive heat transfer is prevalent for the solid components like PEM, GDL, and bipolar plates. It is governed by Fourier's law:

$$q''_{cond} = -\lambda \nabla T \tag{1.8}$$

where λ is the thermal conductivity and ∇T the temperature gradient. Convective heat transfer is critical in thermal management systems employing fluid for heat transfer. It strongly depends on the heat transfer coefficient α, which depends on fluid velocity, wetted surface area, and fluid properties. Radiative heat transfer becomes significant only for HT-PEMFCs operating at elevated stack temperatures due to the increased temperature difference relative to ambient and the corresponding T^4 dependence of radiative heat flux according to:

$$q''_{rad} = \varepsilon\, \sigma_{SB} \left(T_{surf}^4 - T_{\infty}^4\right). \tag{1.9}$$

Here, q''_{rad} is the radiative heat flux, ε is the surface emissivity ($0 \leq \varepsilon \leq 1$), σ_{SB} is the Stefan–Boltzmann constant, T_{surf} is the surface temperature and T_{∞} is the ambient temperature. This equation shows that the radiative heat flux increases strongly with the fourth power of temperature, making it significant at high operating temperatures. Due to the multiple and locally varying heat sources within a PEMFC, an optimized cooling system is required to maintain operation within optimal temperature limits. Consequently, the following chapter provides a comprehensive analysis of conventional cooling methods for PEMFCs.

1.3. Current Thermal Management Practices in PEM Fuel Cells

Proper cooling systems are essential for the effective operation of PEMFCs. Unlike conventional jet engines, where the air flowing through the engine naturally dissipates most of the waste heat, PEMFCs generate significant heat that must be actively managed through dedicated thermal management systems (TMS). Furthermore, PEMFCs operate within relatively narrow temperature windows and require precise thermal control. Considering that for civil aviation the power necessary to operate commercial planes is, e.g., around 20 MW for an A320, highly efficient thermal management systems have to be developed [13]. Cooling systems must be lightweight, compact, and highly efficient, especially in applications with strict weight constraints like aerospace. Additionally, parasitic drag must be minimized to avoid compromising aircraft performance. Two primary types of cooling architectures are common practice. The first is direct cooling, which can be further categorized into open cathode and closed cathode systems. In open cathode systems, the cooling air and reactant air are mixed and supplied directly to the cathode shown in (1) and (2) in Fig. 1.2. In contrast, closed cathode systems use separate air streams for cooling and reactant supply, allowing for better control of airflow and temperature. Advantages of this concept are lower system complexity and reduced mass due to the elimination of external active cooling systems. Certain challenges include the reduced density of the incoming air, which first has to be compressed in order to provide enough oxygen particles for the cathode reaction, resulting in a higher temperature of the cooling air stream. Intercoolers become necessary for LT-PEMFC to ensure the temperature window required for profound performance is kept [25]. Due to the additional friction, the already existing parasitic drag caused by the freestream air used for the cooling system rises further. Additionally, the cooling performance, quantified by the convective heat transfer rate, is given by:

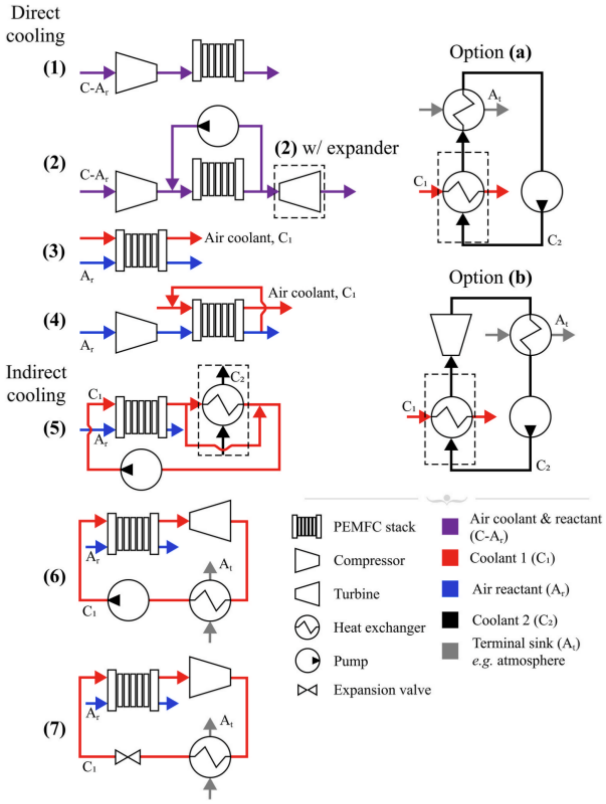

Figure 1.2: Thermal management cooling architectures [25]

$$\dot{Q} = \alpha A (T_{FC,\mathrm{surf}} - T_\infty) \tag{1.10}$$

where α is the heat transfer coefficient, A is the wetted area, T_∞ is the free-stream air temperature, and $T_{FC,\mathrm{surf}}$ is the surface temperature of the fuel cell system. The cooling performance is strongly dependent on the temperature difference between the fuel cell and the incoming air. Therefore, depending on the flight mission, the performance of the cooling system may strongly depend on atmospheric conditions, thus reducing the reliability of the cooling system in an environment that is very restrictive in terms of certification of new propulsion systems. For the concept of the open-cathode system, another arising challenge is that the air at high altitudes is fairly dry, posing a challenge to the functionality of the cathode, especially for LT-PEMFCs, as stated in the previous chapter. In addition, due to the low thermal conductivity, air-cooled FC stacks are commonly suitable for stacks with a thermal output of < 5 kW [84]. The alternative cooling principle is indirect cooling. In this concept, cooling is primarily established via liquid cooling circuits that transfer heat to a heat exchanger or radiator, as depicted in (5), (6), and (7) of Fig. 1.2. The main disadvantages are increased system complexity, parasitic power drawn from the fuel cell output to pump the coolant and maintain low temperatures, and increased weight. Despite these drawbacks, using a liquid as a coolant offers a higher maximum heat transfer capability due to the higher heat transfer coefficients of liquids compared to gases, and generally allows fully independent control of the coolant and reactant flows. Heat exchangers that transfer heat from the stack first into a liquid coolant and subsequently into cooling air combine a liquid-based internal cooling approach with an air-based external heat rejection method. Since the stack is never in direct contact with the air stream, this architecture is still classified as an indirect cooling system, even though ambient air ultimately serves as the heat sink. The use of a liquid coolant enables accurate temperature control within the stack, while the heat exchanger uses the freestream air to ensure effective heat rejection to the environment. This configuration is often viewed as a compromise, as it retains the thermal stability and high heat transfer capability of liquid cooling while benefiting from the simplicity of convective heat rejection to ambient air. However, these advantages come at the cost of increased system complexity and parasitic power demand for pumps and, if required, compressors to maintain coolant flow and adequate air mass flow [12].

In this context, pulsating heat pipes (PHPs) emerge as a promising alternative to mitigate the aforementioned complexities of conventional PEMFC cooling approaches, while maintaining efficient heat transfer.

1.4. Fundamentals of Pulsating Heat Pipes

A PHP is a passive cooling system that does not require any pumps or parasitic power to operate. In this sense, PHPs are self-regulating systems that automatically respond to the applied heat input. However, they only operate reliably within specific heat flux and temperature windows, outside of which stable oscillatory behavior cannot be sustained. Fig. 1.3 illustrates the basic design of a pulsating heat pipe. It consists of a single capillary tube that is meandered into multiple turns and divided into three main sections: an evaporator, a condenser, and an optional adiabatic section. In the manufacturing process, the PHP is first evacuated well below atmospheric pressure. In some experimental setups, the pressure prior to the filling procedure is reduced to values below 10^{-4} Pa, as reported for example in experiment II of the dissertation by S. Khandekar [34]. After evacuation, the filling process is initiated using different working fluids and varying fill ratios (FR), defined as the ratio of liquid volume to total internal volume of the PHP. Typical working fluids include water, ethanol, and methanol, among others, depending on the targeted operating temperature range and application. The choice of working fluid and FR has a significant influence on the start-up behavior and overall thermal performance of the PHP. The formation of liquid slugs and vapor plugs inside a capillary tube is primarily

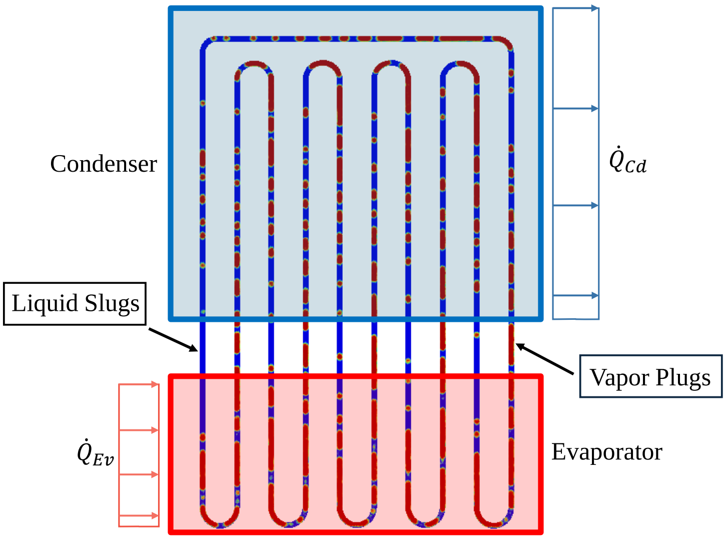

Figure 1.3: Schematic representation of a PHP

governed by capillary forces, which can be characterized using the Bond number:

$$\mathrm{Bo} = \frac{(\rho_l - \rho_v)\, g\, D_h^2}{\sigma}. \tag{1.11}$$

Here, ρ_l and ρ_v denote the densities of the liquid and vapor phases, respectively, g is the gravitational acceleration, D_h is the hydraulic tube diameter, and σ is the surface tension of the fluid. The Bond number represents the ratio between gravitational forces and capillary forces. For Bond numbers significantly smaller than one, capillary forces dominate over gravitational forces. For a suitable tube diameter, this dominance leads to the spontaneous formation of alternating liquid slugs and vapor plugs within the tube. Although the literature is not entirely consistent regarding whether the liquid or vapor phase should be referred to as a slug or a plug, the existence of these discrete phase structures is essential for proper PHP operation [54, pp. 63-139]. During operation, the working fluid must reach a minimum temperature to initiate evaporation. Once this threshold is exceeded, the saturation pressure inside the vapor plugs increases with rising temperature. In the evaporator section, the liquid films surrounding the vapor plugs begin to evaporate, causing a local pressure rise. This pressure increase drives the adjacent liquid slugs and vapor plugs toward the condenser. In the condenser section, which acts as a heat sink, the vapor condenses and releases its latent heat to the surroundings. The resulting condensation leads to a pressure decrease, thereby sustaining the oscillatory motion of the working fluid. This continuous sequence of evaporation and condensation gives rise to the characteristic pulsating or oscillating behavior from which the pulsating heat pipe derives its name. The process is inherently transient and strongly dependent on the applied heat input. Although the geometric design and basic operating principle of a pulsating heat pipe are relatively simple, its behavior is difficult to predict accurately. This is due to the complex interaction of multiphase flow, non-stationary thermodynamic processes, and strong coupling between heat transfer and fluid dynamics.

Pulsating heat pipes can be arranged at different angles relative to the horizon. Due to the connected meander structure, they are able to operate in multiple orientations, although some orientations perform better than others. Under gravitational conditions, a vertical orientation is generally considered optimal for maximizing heat transfer performance across different FR configurations, although the optimal performance depends on the combination of several parameters, including FR, heat input, working fluid, and geometry [48]. Furthermore, in vertical operation, gravity assists the condensed fluid in flowing back to the evaporator section. Thus, in environments with reduced gravity or no gravity at all, the orientation becomes less significant. For space or non-gravitational conditions, it is interesting to note that due to the Bond number approaching zero, the critical pipe radius required to establish liquid slugs and vapor plugs increases. This leads to a higher attainable volumetric flow rate and therefore an increased heat transport capability [27]. Various combinations of the mentioned parameters defining a PHP lead to different nonlinear system responses, resulting in variable outcomes not only in terms of optimal heat conduction and transfer, but also in the oscillation behavior and flow regimes inside a PHP. As shown in Fig. 1.4, slug flow, semi-annular flow, and annular flow patterns can be distinguished as long as no dryout occurs. Dryout describes the complete evaporation of the liquid phase such that only vapor remains inside the pipe. The flow behavior inside the PHP can further differ depending on temperature variations and local temperature peaks or drops along the pipe. Therefore, predicting flow patterns and deriving time-dependent heat transfer behavior is highly complex and inherently subject to uncertainties. The established metric to evaluate overall PHP performance is the thermal resistance:

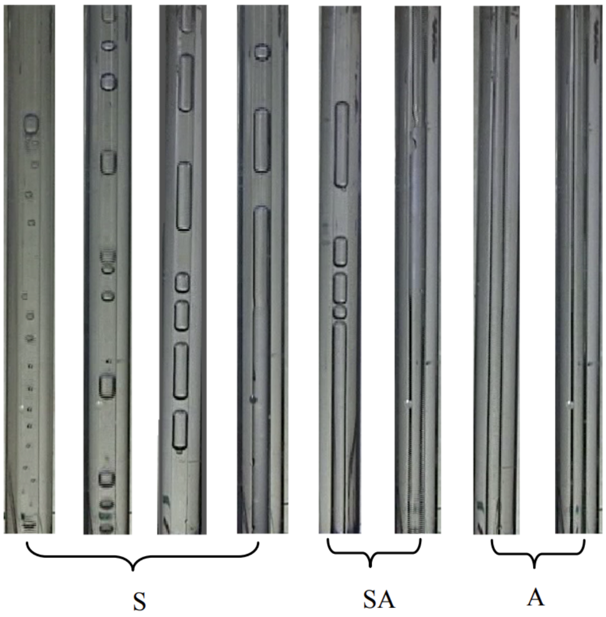

Figure 1.4: Flow regimes of PHPs [41]

$$R_{th} = \frac{T_{ev} - T_{cd}}{\dot{Q}} \tag{1.12}$$

where T_{ev} and T_{cd} denote the average temperatures of the evaporator and condenser sections, respectively, and $\dot{Q}$ is the heat input at the evaporator section. By inverting the thermal resistance, the equivalent thermal conductivity can be defined as:

$$\lambda_{eq} = \frac{L_{cd-ev}}{A_{cr}} \frac{1}{R_{th}}. \tag{1.13}$$

Here, L_{cd-ev} represents the distance between the condenser and evaporator sections, and A_{cr} denotes the cross-sectional area of the PHP [45]. These values are determined under quasi-stationary conditions after a constant heat input has been applied to the evaporator for a sufficiently long period, such that the temperatures, averaged over the pulsation cycles, can be calculated. Several thermophysical processes govern the heat transfer mechanisms in PHPs, which can be decomposed into sensible and latent heat contributions. A local energy balance at the fluid–vapor interface yields:

$$\dot{Q}_i = \dot{Q}_{cond,i} + (\dot{m}_{ji} - \dot{m}_{ij})h_i \tag{1.14}$$

where $\dot{Q}_{cond,i}$ denotes the conductive heat transfer across the interface, typically described by Fourier's law (Eq. 1.8), and $\dot{m}_{ij}$ represents the mass transfer rate between phases i and j [2]. The term h_i is the specific enthalpy of phase i. Sensible heat refers to the portion of energy associated with temperature variations within a phase without phase change. For a flowing liquid stream, the sensible heat rate can be written as:

$$\dot{Q}_{\text{sensible}} = \dot{m}_l c_{p,l}(T_l - T_{\text{ref}}). \tag{1.15}$$

In this expression, $\dot{m}_l$ is the liquid mass flow rate, $c_{p,l}$ is the specific heat capacity, T_l is the liquid temperature, and T_{ref} is a reference temperature. Latent heat represents the energy associated with phase change. During evaporation, the liquid absorbs the latent heat of vaporization, which is subsequently transported by the vapor phase in addition to its sensible enthalpy. The specific enthalpy of the vapor can be expressed as:

$$h_v = c_{p,v}(T_v - T_{\text{ref}}) + h_{lv} \tag{1.16}$$

where $c_{p,v}$ is the vapor heat capacity and h_{lv} is the latent heat of vaporization. Thus, the total enthalpy of the vapor phase consists of both sensible and latent contributions relative to a reference state. Furthermore, the operational behavior can be divided into distinct, heat-input-dependent regimes with time-dependent characteristics. The initial regime is pure conduction, which occurs during start-up prior to the onset of fluid motion, where heat transfer is governed solely by conduction. With increasing heat input, a pulsation regime is established, characterized by local pressure fluctuations that induce small-scale fluid motion. This is followed by the start-up regime, in which oscillatory flow structures become increasingly pronounced. At higher heat inputs, a fully developed activation regime is reached, characterized by enhanced heat transport mechanisms and a reduction in evaporator temperature. Eventually, dryout occurs when the liquid inventory is largely depleted, leading to a corresponding increase in thermal resistance.

After their initial development and patent in 1990 by Akachi Hisateru [5], pulsating heat pipes have been widely investigated and applied in various fields, including rotating machinery, battery thermal management, electronic cooling, and aerospace applications due to their high heat transfer capability, adaptability to different orientations, and reliability. Experimental studies demonstrate that PHPs can effectively enhance thermal performance under centrifugal forces, non-uniform heating, and microgravity, making them suitable for both terrestrial and space applications. Advanced designs such as three-dimensional, flat-plate, and hybrid PHPs further improve start-up behavior, heat transport capacity, and thermal uniformity [22]. For integration into the thermal management of fuel cells, pulsating heat pipes offer a practical and efficient method of heat transfer. The structure of bipolar plates allows for spaces in which PHPs can be arranged. The membrane electrode assembly (MEA), which is the main heat source, is surrounded by bipolar plates, which are used to distribute hydrogen and oxygen across the MEA. Inside the welded plates, space is provided for the pulsating heat pipes, replacing conventional cooling channels that depend on the parasitic power of pumps and a coolant reservoir, with interconnected meandered tubes, as shown in Fig. 1.5. The evaporator section of the PHP will be inside the MEA region. The condenser section could be realized by extending the PHP structure beyond the stack, for example, into oncoming airflow as a heat sink, conceptually shown in Fig. 1.6. Results of numerical simulations of this assembly under certain simplifications by Franke et al. imply feasibility for PHPs integrated within the bipolar plates of HT-PEMFCs in certain operating windows [24]. This is based on the achieved thermal resistance of the modeled pipe strand, and thus the transferred heat without exceeding an operating temperature limit of 473.15 K for HT-PEMFCs and staying within a 10-degree temperature difference between evaporator and condenser. This simplified numerical approach indicates that integration of pulsating heat pipes into fuel cells via bipolar plates is theoretically feasible and can provide sufficient heat transfer performance.

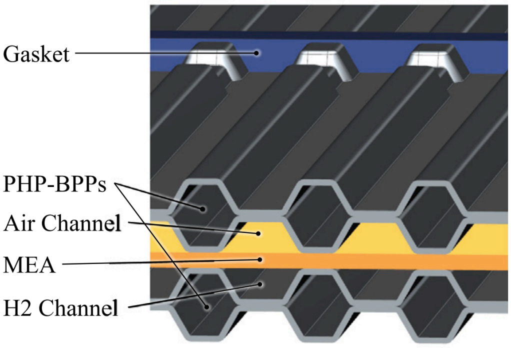

Figure 1.5: Cross section of MEA-assembly with two PHP-BPPs and gasket [24]

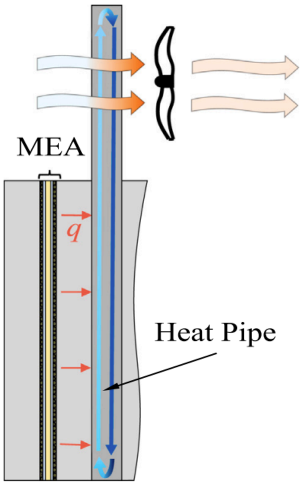

Figure 1.6: Heat pipe assisted air cooling [24]

1.5. Modeling Approaches for Pulsating Heat Pipe Performance

To enable the further assessment and optimization of PHP designs, reliable performance prediction models are required.

Several approaches already exist for predicting the heat transfer of PHPs. These mainly consist of experiments on entire PHP systems, with some focusing on the specific thermophysical phenomena driving heat transfer in PHPs. Although there are approaches using simplified numerical simulations, for example, to one or two dimensions that work in narrow operating regimes and geometries [53], entire 3D simulations still lack reliability and suffer from great computational effort due to the complex underlying physics involved.

A method of significantly reducing computational effort compared to numerical simulations is using machine learning (ML) to study patterns of PHP behavior. However, for an ML model to learn the underlying physical correlations, a substantial amount of training data is required. Literature suggests that machine learning approaches, including neural networks (NNs) and gradient boosting models, enable the prediction of thermal resistance in PHPs with reasonable accuracy. However, the majority of these methods remain case-specific, often tailored to limited datasets and particular working fluids, thereby constraining their predictive capabilities [81, 60]. For instance, studies focus primarily on a single working fluid alongside geometrical parameters, yielding promising outcomes within those narrow scopes [4]. In contrast, other investigations have developed NN models based on large datasets aggregated from literature, such as one using over sixteen hundred entries and another employing more than seven hundred data points, incorporating both geometrical and dimensionless parameters, and demonstrated effective predictive capabilities for various working fluids and operational conditions in PHPs [62, 80].

2. Methodology

This chapter presents the overall methodological approach of this work, which combines a comprehensive dataset derived from published experimental studies on PHPs with a machine learning framework based on gradient boosting to predict their thermal performance. In addition, a numerical approach is developed as a complementary method to provide deeper insights into the underlying mechanisms of PHP operation, contributing to a better understanding of the complexity of the involved physical processes and the associated challenges in their simulation. In both approaches, thermal resistance is used as the central performance metric, serving as a consistent parameter for evaluating and comparing the results with experimental data for validation.

To construct a more versatile ML model capable of forecasting thermal resistances across diverse parameter variations, geometries, working fluids, and mixtures, thereby supporting iterative design processes, this work compiles a comprehensive dataset comprising 3,473 data points from experimental publications spanning from 2004 to 2025 (listed in Appendix B.1-B.2). The developed model follows a multiple-input, single-output structure, where the input features describe the variables whose interactions and combined influence on the output are to be learned by the model, and the target variable to be predicted is the thermal resistance. A subset of input features consists of dimensionless parameters chosen to represent the underlying physics and improve model generalization compared to raw variables. Central geometrical properties were also introduced. Given the wide range of geometrical configurations, sizes, and operating conditions of PHPs, parameters that consistently appear in multiple studies and are known to influence performance were selected as key input features. This approach combines dimensionless physical and geometrical parameters, resulting in a dataset that spans diverse geometries as well as various working fluids and mixtures. The final parameter set is presented and discussed in detail in a later section. The machine learning model further incorporates extensive hyperparameter optimization, robust cross-validation, and additional optimization techniques, which are described in Section 4.2. Additionally, a Shapley Additive exPlanations (SHAP) analysis is employed to interpret the final model by quantifying the contribution of each feature to the predictions and identifying underlying feature interactions, thereby enhancing the transparency and interpretability of the model.

The numerical approach considers two different PHP geometries for which experimental results are available and well documented, serving as a basis for comparison. Computational meshes are generated for both geometries, and the core models are selected and configured to capture the complex multiphase thermophysical processes within the PHP. A grid independence study is conducted, and the finest mesh that remains computationally feasible is selected, taking into account the high computational cost and complexity associated with PHP simulations. Appropriate boundary conditions are defined, and the final results are analyzed and compared with both the experimental measurements and the predictions of the machine learning model for the simulated cases.

3. Numerical Simulation of Pulsating Heat Pipes

Numerical simulations of PHPs are computationally demanding due to the complexity of the underlying physics and the long startup times involved. They are therefore mostly carried out under various constraints and simplifications, in order to reduce simulation time and model complexity. To validate and compare the results of the approach presented in this work using computational fluid dynamics (CFD) simulations, the dissertation by S. Khandekar is selected as a reference, namely experiments II and III, which feature two different PHP designs [34]. All simulations were conducted using ANSYS Fluent 2024 R1.

3.1. Simulation Setup

The setup of CFD simulations is crucial for obtaining accurate and physically reliable results while maintaining a feasible computational cost. A key aspect in this context is the discretization of the computational domain, which is defined by the selected mesh. The mesh quality, resolution, and suitability for the underlying flow physics strongly influence the overall simulation performance. While a finer mesh generally improves accuracy, it also leads to significantly increased computational effort and simulation time. The selection of the mesh is significant for the overall performance of the simulation.

3.1.1. Mesh Generation and Selection

The simulation was performed in three dimensions, but to reduce computational time, a symmetry plane was introduced through the pipes to halve the domain as seen in Fig. 3.1. Generally, different mesh densities are tested to identify the point at which the simulation accuracy is no longer mesh-dependent (more details in Section 3.2) for a simplified case to reduce computation time.

In addition to the overall resolution of the geometry used in the simulation, the mesh should also achieve a certain quality measured by multiple quality metrics describing the geometrical relationships between cells within the mesh. For all meshes considered in the conducted mesh study in section 3.2, except for the smallest cell counts in both experiments (100k cells for experiment III and 160k cells for experiment II, which slightly exceeded some of the limits regarding good mesh quality), the maximum skewness is kept below 0.9. The average skewness is lower than 0.25, which falls into the excellent cell quality range [19]. The skewness measures the distortion of a cell relative to its optimal

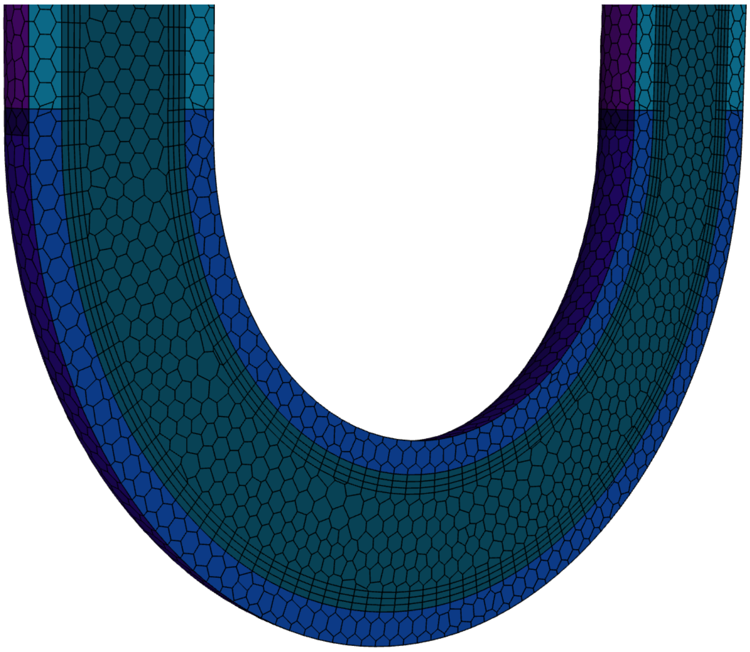

Figure 3.1: Computational mesh of the halved fluid domain utilizing a symmetry plane for the numerical replication of experiment II (700,000 Cells)

geometry, and low skewness values help maintain numerical stability. Furthermore, the orthogonal quality is maintained above 0.2 for the worst cell, while the average orthogonal quality is kept above 0.9, placing it in the excellent category of cell quality. In some regions, especially in the meander structures of experiment III, the mesh had to be refined locally to ensure good mesh quality. The minimum orthogonal quality and maximum skewness values remain well above the warning thresholds and comfortably below the error thresholds prescribed in the ANSYS meshing user's guide, while the maximum aspect ratio slightly exceeds its warning limit yet stays substantially below the corresponding error limit across all evaluated meshes [7].

In mesh consideration for PHPs, the resolution of the fluid close to the wall, for example, to resolve the interfaces between liquid slugs and vapor plugs, and the contact angles between fluid and the wall, can significantly improve simulation accuracy for heat transfer in PHPs. However, for three-dimensional simulations of PHPs, resolving these layers requires a fine mesh and therefore a great amount of computational power. Additionally, since boundary-layer heat transfer is not the dominant heat transfer mechanism in PHPs, such vastly resolved meshes are computationally impractical [55].

The thermal boundary layer thickness may differ from the velocity boundary layer depending on the fluid's Prandtl number. For fluids with high Prandtl numbers (e.g., water-based), the thermal layer is thinner than the velocity layer, which could require a finer mesh close to the wall to fully resolve the temperature gradient [38]. However, in PHPs, phase change dominates heat transfer rather than conduction through boundary layers, so the selected mesh is considered adequate for capturing the primary thermal behavior under the computational possibilities available [31]. As described in the dissertation of M. Opalski, the computational focus for simulating PHP accurately lies instead on accurately modeling the interface sharpness [55]. This is essential for correctly computing the phase interactions, which are the dominant mechanisms governing heat transfer and operation in PHPs. To provide a benchmark for the approximate simulation time of the final simulations, the mesh selected for experiment II (see Section 3.2) required approximately 14 days of computation on an AMD EPYC 7702 (48 cores) to simulate about 30 seconds of physical time. At that stage, the simulation had not yet fulfilled multiple oscillations, making the calculation of an averaged thermal resistance impossible. For the majority of the conducted simulations, at least three weeks of computation were required before multiple oscillations were captured, enabling a reliable calculation of thermal resistance. This highlights the substantial computational effort required for PHP simulations.

3.1.2. Thermophysical Multiphase Modeling

Pulsating heat pipes underlie complex thermophysical principles defined by the dynamic interplay between two fluid phases. The phenomena are described by the Navier-Stokes equations of fluid dynamics, which form a system of coupled partial differential equations describing mass conservation, momentum, and energy. The conservation of mass for a Newtonian fluid is given by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0. \tag{3.1}$$

Here, ρ is the density, t is the time, and $\mathbf{u}$ is the velocity field. The incompressible momentum conservation equation is:

$$\rho \left(\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \right) = -\nabla p + \mu \nabla^2 \vec{u} + \rho \vec{g} \tag{3.2}$$

and the conservation of energy is described by:

$$\rho \left(\frac{\partial e}{\partial t} + \vec{u} \cdot \nabla e \right) = \Phi + \nabla \cdot (\lambda \nabla T) \tag{3.3}$$

where μ is the viscosity and Φ the dissipation function of mechanical energy [42, pp. 22-23]. These equations only apply to a continuous fluid. However, since a system that contains two fluid phases is investigated, both the underlying equations and the CFD setup must be extended accordingly. For sharp interfaces and the motion of bubbles and slugs, the Volume-of-Fluid (VOF) model is recommended by ANSYS and has been a common choice in previous CFD studies of PHPs [56, 8]. The VOF model solves a single set of momentum equations while tracking the fluids throughout the domain. The interface between phases is tracked via a modified version of continuity Eq. 3.1 for the q^{th} phase:

$$\frac{1}{\rho_q} \left[\frac{\partial}{\partial t} (\alpha_q \rho_q) + \nabla \cdot (\alpha_q \rho_q \vec{u}_q) \right] = S_{\alpha_q} + \sum_{p=1}^{n} (\dot{m}_{pq} - \dot{m}_{qp}) \tag{3.4}$$

where α_q is the volume fraction of phase q, ρ_q the corresponding density, u_q the velocity of phase, $\dot{m}_{pq}$ the mass transfer from phase p to q and S_{α_q} a source term (default $S_{\alpha_q} = 0$). The volume fraction of the primary phase is computed using the constraint:

$$\sum_{q=1}^{n} \alpha_q = 1. \tag{3.5}$$

The discretization of this equation is explicit in this approach. The momentum equation is solved throughout the domain, and the resulting velocity field is shared among all phases:

$$\frac{\partial}{\partial t}(\rho\vec{u}) + \nabla \cdot (\rho\vec{u}\vec{u}) = -\nabla p + \nabla \cdot \mu \left[\left(\nabla\vec{u} + \nabla\vec{u}^T\right) - \frac{2}{3}\nabla \cdot \vec{u}I\right] + \rho\vec{g} + \vec{F} \tag{3.6}$$

where the phase-dependent properties ρ and μ are computed from the volume fractions of all phases and $\vec{F}$ describing additional body forces. Similarly, a single energy equation is solved for all phases:

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\vec{u}(\rho e + p)) = \nabla \cdot \left(\lambda\nabla T - \sum_q \sum_j h_{j,q}\vec{J}_{j,q} + (\tau \cdot \vec{u})\right) + S_h \tag{3.7}$$

with mass-averaged energy:

$$e = \frac{\sum_{q=1}^{n} \alpha_q \rho_q e_q}{\sum_{q=1}^{n} \alpha_q \rho_q}. \tag{3.8}$$

The temperature T is shared among all phases. The right-hand side terms represent energy transfer due to heat conduction $\lambda\nabla T$, species diffusion $h_{j,q}J_{j,q}$, and viscous dissipation $\tau \cdot u$, respectively. The source term S_h includes volumetric heat contributions. The viscous dissipation term is neglected in the energy equation, as the liquid phase is assumed to be incompressible, and the Brinkman number:

$$Br = \frac{\mu u^2}{\lambda\,\Delta T} \tag{3.9}$$

is much smaller than unity ($Br \ll 1$) for the operating conditions considered in this work. This holds for both the liquid phase and the gas phase, the latter being modeled as an ideal gas. Consequently, viscous heating effects are negligible [9].

Since phase change constitutes the primary driving mechanism of PHP operation, the rates of evaporation and condensation must be modeled with sufficient accuracy. To this end, a user-defined Antoine equation is implemented to provide a pressure-dependent saturation temperature, based on data from the National Institute of Standards and Technology (NIST) [36]. Furthermore, a correlation for the strongly temperature-dependent viscosity of the liquid phase of ethanol is used based on the work of Sotiriadou et al., as ethanol is the working fluid simulated in all simulations [73]. Since the temperature dependencies of the remaining properties of the liquid phase are lower, they are neglected. The vapor phase is modeled as an ideal gas to accurately capture pressure fluctuations, which are essential for the pulsation behavior. Vapor-phase properties such as viscosity and thermal conductivity are assumed constant due to their comparatively low values and limited influence relative to the liquid phase.

Furthermore, the relaxation coefficients of the Lee model, used to model the rates of evaporation and condensation, have been selected in accordance with the values reported by Opalski et al. [56], showing decent results using ethanol as a working fluid, namely $\beta_c = 10$ for condensation and $\beta_v = 0.05$ for evaporation. Within ANSYS Fluent, the Lee model serves as the standard approach for modeling the intensity of phase change processes in the VOF framework [6]. However, the model coefficients require careful tuning to match the physical behavior of the specific working fluid and operating conditions.

As already mentioned, the modeling of the phase interfaces is a crucial part of the PHP simulation. There are, in principle, two methods of doing so. One is a geometrical reconstruction of the interface based upon linear slopes within each cell near the interface. This is the most accurate, but it also comes with the highest computational expense. As an alternative that requires less computational expense at the cost of lower accuracy of the interface formulation, the compressive interface capturing scheme can be used [78]. After extensive iterations and testing with these setups, the compressive interface method was not able to build sharp interfaces and smeared the liquid and vapor phases. The geometric reconstruction, on the other hand, calculated sharp interfaces without any smearing, justifying its selection. Another significant parameter in CFD is the Courant-Friedrichs-Lewy (CFL) number, defined as:

$$\mathrm{CFL} = \frac{u\Delta t}{\Delta x} \tag{3.10}$$

where u is the local flow velocity, Δt is the time step size and Δx is the local cell size. The CFL number characterizes how far information travels within a cell or the mesh per timestep. The ANSYS default CFL number limit for the explicit VOF model is 0.25, while the maximum allowable CFL number was set to 1. According to the stability condition of the CFL number, information should not travel more than one computational cell per time step [39, p. 70]. Therefore, adaptive time stepping was employed, in which the time step size was automatically adjusted based on the CFL number to ensure numerical stability. The Pressure-Implicit with Splitting of Operators (PISO) pressure-velocity coupling scheme was used, as it is suggested by ANSYS for transient simulations. Further solution methods were kept at the ANSYS default settings for PISO pressure-velocity coupling [20].

For the present simulated cases, ethanol in capillary channels with an inner diameter of $D = 2$ mm is considered. The temperature range considered spans from ambient conditions ($T_{min} = 298.15$ K) up to approximately $T_{max} = 353$ K. The characteristic Reynolds number is estimated as:

$$\mathrm{Re} = \frac{\rho\, u\, D}{\mu} \tag{3.11}$$

with ρ denoting the fluid density, μ the dynamic viscosity, and u the characteristic oscillation velocity of liquid slugs and vapor plugs. Typical velocities reported for PHPs in related operating regimes lie in the range of 0.1–1 m/s [63] with the mean velocity remaining below 0.5 m/s. The corresponding expected average Reynolds number is $Re \approx 1200$. Since the Reynolds number is well below the critical threshold for transition to turbulent flow ($Re \approx 2300$), laminar flow is assumed [70]. An additional comparison was conducted using the $k-\epsilon$ turbulence model, which was applied in previous PHP simulation studies, to assess its impact on an exemplary simulation and validate the laminar flow assumption [17]. The results showed only a minor influence on the solution. The comparing plots are shown in Appendix A.1. The $k-\epsilon$ turbulence model was activated after a flow time of 14.44 s. From that point onward, the relative difference between the laminar and turbulent simulations in terms of the time-averaged thermal resistance was approximately 2.53%.

Notably, the laminar treatment showed slightly better agreement with the experimental data.

In addition to selecting appropriate physical models, the definition of suitable boundary conditions is a crucial step to accurately represent the experimental setups. These include, for example, temperature and pressure conditions as well as other relevant constraints. Before proceeding with the grid studies to determine the most efficient meshes, the following section outlines and discusses the boundary conditions and simplifications adopted in this work.

3.1.3. Boundary Conditions

The greatest challenge encountered while setting up the simulation is that, due to model complexity, the simulation needs a lot of computational time to produce significant results, especially considering PHPs have start-up times reaching up to multiple minutes [34]. The first step was to perform a simulation with the singular purpose of determining the initial vapor plug and liquid slug formation and distribution. For the initial conditions of the working fluids, the temperature is set to a value of 298.15 K for both experiments. For consistency and to achieve the specific fill ratio, the pressure is set to the saturation pressure of 7900 Pa corresponding to the initial temperature. This limits phase reactions and instabilities. Another important parameter is the contact angle of liquid slugs to the pipe. It determines the wetting of the surface and the plug transport capability of the pipe. For copper and glass as the used materials in the two experiments, the contact angles are based on experimental tables suggested in literature [15, 21]. Hence, the ethanol-copper contact angle with respect to vapor as the surrounding phase was set to 11° and 1° for glass surfaces, respectively. The surface of the pipes is considered smooth. The fluid used in the simulations was ethanol with predefined characteristics. In addition to the user-defined functions used to calculate the saturation temperature and the viscosity of the liquid phase, the density of the vapor phase is set to obey the ideal gas law, while the density of the liquid phase is set to be constant. Further characteristics of the fluid, for example, the thermal conductivity, are also set to constant values and not defined as temperature- or pressure-dependent in order to save calculation time for both phases.

The simulation was initialized with the aspired volumetric fraction of the liquid and vapor phase of the working fluid corresponding to the experiments. No heat sources or sinks were applied at this stage.

The first method of validating the simulation is the comparison with the aforementioned experiments executed by Khandekar [34]. It is investigated if the distributions match the experimental result (Appendix A.2) and if the liquid-vapor interfaces, e.g., seen in Fig. 3.2, are resolved sharply. The fluid distribution is considered after reaching a stationary volume-averaged static pressure. This distribution is interpolated into the subsequent PHP simulations, in which the heat sources and sinks are applied according to the experimental conditions. This approach ensures a consistent initial state across all simulations, thereby enhancing the comparability of the results and minimizing influence from tran-

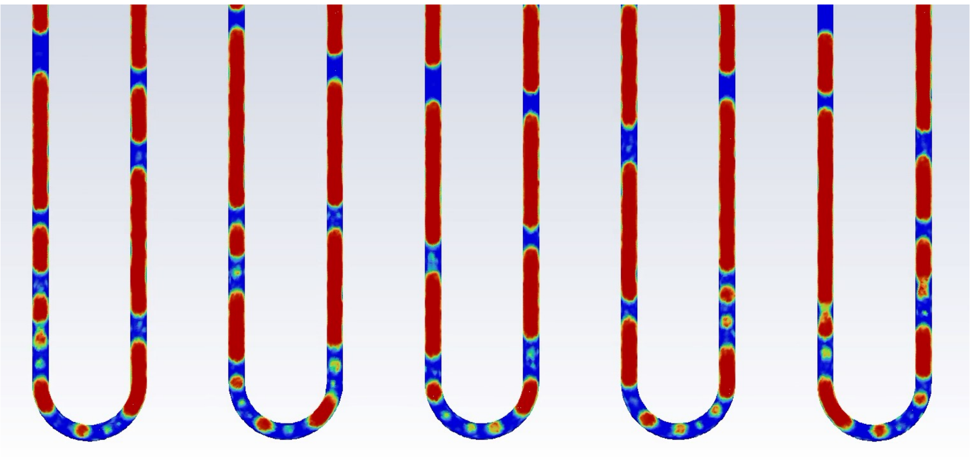

Figure 3.2: Representative liquid-vapor distribution and interface capturing under active heat input

sient startup effects.

Boundary Condition of Evaporator and Condenser Sections

For the simulations, a Neumann boundary condition shown in Eq. 3.12, is used to model the heat source at the evaporator.

$$q'' = -\lambda \frac{\partial T}{\partial n} \tag{3.12}$$

Because the efficiency of the heat sink, and thus the heat flux, depends on the temperature of the fluid within the pipe, a constant heat flux boundary condition at the condenser would not be appropriate. Therefore, a Dirichlet boundary condition in the form of a fixed temperature is applied to the outer surface of the condenser, corresponding to the experimental measurements, as this approach is more representative of the actual experimental conditions. That approach was used, e.g., in a 2-dimensional simulation conducted by Opalski et al. [56], whose results showed some compliance with experimental results in a 2-dimensional PHP simulation.

Heat Loss to the Environment

In order to justify the assumption of an adiabatic boundary condition of the remaining surfaces for both experiments, an analytical calculation of the theoretical heat loss in the remaining pipe sections was conducted, corresponding to a maximum heat input of 74.7 W used in experiment III (experimental setup illustrated in the Appendix A.3). This heat input results in a specific heat flux of $q'' = 198.85$ kW/m^2 through the surface of the evaporator. In the experiment, the measured average surface temperature T_w of the evaporator was approximately 100 °C for this heat input. Thermocouples on the adiabatic section of the pipe measured temperatures of 60 °C and 50 °C for the left and right pipe strands, respectively.

For the copper sections in experiment III, the heat losses due to natural convection are evaluated. Assuming an ambient air temperature of approximately 25 °C, consistent with the experimental conditions and adopting a conservatively assumed pipe surface temperature of 100 °C, the Grashof number for a horizontal cylindrical surface in natural convection is calculated according to [38]:

$$Gr_d = \frac{g\,\beta\,\Delta T\,D^3}{\nu^2}. \tag{3.13}$$

Here, $g = 9.81$ m/s^2 is the gravitational constant, β is the thermal expansion coefficient estimated as the inverse of the average film temperature (surface and ambient), D is the outer tube diameter, and ΔT denotes the temperature difference between surface and ambient air. Using the corresponding material properties of air, this expression yields a Grashof number of approximately $Gr_d \approx 2.77 \times 10^2$. With the Prandtl number of air at 25 °C being $\mathrm{Pr} \approx 0.7$, the resulting Rayleigh number becomes

$$Ra = Gr_d\,Pr \approx 2 \times 10^2. \tag{3.14}$$

For natural convection around horizontal cylinders, the commonly used empirical correlations are given by:

$$\text{laminar:} \qquad \overline{\mathrm{Nu}}_d = 0.53\,(\mathrm{Gr}_d\mathrm{Pr})^{1/4} \qquad \text{for} \quad 10^4 < \mathrm{Gr}_d\mathrm{Pr} < 10^9, \tag{3.15}$$

$$\text{turbulent:} \qquad \overline{\mathrm{Nu}}_d = 0.13\,(\mathrm{Gr}_d\mathrm{Pr})^{1/3} \qquad \text{for} \quad 10^9 < \mathrm{Gr}_d\mathrm{Pr} < 10^{12}. \tag{3.16}$$

For Rayleigh numbers smaller than 10^3, these correlations are no longer valid. In this regime, the convective heat transfer is extremely weak, and for $Ra < 300$ convection becomes negligible, and the Nusselt number approaches unity [28]. Since the thermal conductivity of air is low, the corresponding heat loss is negligible as well. Assuming $\mathrm{Nu} = 1$ and rearranging the definition:

$$\mathrm{Nu} = \frac{\alpha D}{\lambda} \tag{3.17}$$

the heat transfer coefficient becomes

$$\alpha \approx 8.67\ \mathrm{W/(m^2K)}.$$

The convective heat loss can then be estimated by:

$$\dot{Q} = \alpha\,A\,(T_w - T_\infty). \tag{3.18}$$

With $A \approx 1.7 \cdot 10^{-3}$ m^2, the resulting heat loss is approximately $\dot{Q} \approx 1$ W for an assumed surface temperature of 100 °C applied across the entire lower copper section between the evaporator and the glass tubes. As this temperature is significantly higher than the actual measurements, the heat loss is neglected. Consequently, the boundary conditions in the simulations are set to be adiabatic for both the remaining copper and glass sections.

For experiment II, illustrated in the Appendix A.4, the pipe domain consists exclusively of copper. The pipe sections that are not part of either evaporator or condenser sections are negligible in heat loss and, therefore, set as adiabatic, as they are limited to the meanders located below the heated section of the evaporator and insulated in the experiment.

3.2. Grid Independence Study

Before the final simulations were conducted, a grid study was performed to identify the point at which further mesh refinement no longer significantly improves simulation accuracy, thereby enabling the selection of an optimal trade-off between accuracy and computational cost.

Due to the high computational cost of the simulations and limited available resources, the grid study was performed using a simplified representation of the final PHP simulations. Instead of modeling the full thermofluidic behavior, only the fluid distribution and the corresponding static pressure evolution were considered. Transient simulations were carried out to determine the quasi-stationary behavior of the system. To avoid convergence issues associated with the initial formation of plugs and slugs, the physical model was further simplified. The heat input was set to zero, the saturation temperature was fixed at 298.15 K corresponding to a pressure of 7900 Pa, and the gas density was assumed to be constant.

Meshes that demonstrated fast convergence and low variance were selected for both experiments. These meshes will be used in the final PHP simulations. The convergence was assessed following the statistical convergence assessment (SCA) methodology proposed by Scandurra et al. [69]. The essence of the method is first to evaluate the cutoff to a steady-state. A reverse root mean square error (RMSE) was first computed in order to identify the end of the transient phase:

$$RMSE_i = \frac{\sigma_{sd}(x_i, x_{i+1}, ..., x_N)}{\sqrt{N - i + 1}} \tag{3.19}$$

where σ_{sd} denotes the standard deviation of the remaining signal from iteration i to N. A smoothing procedure was applied to the RMSE curve to reduce high-frequency oscillations, and the iteration corresponding to the minimum filtered RMSE was selected as the beginning of the steady-state region. The steady-state region is then evaluated based on two convergence conditions that have to be satisfied simultaneously.

(i) The confidence interval of the mean was smaller than a user-defined tolerance, and

(ii) the total variation induced by the regression slope over the steady-state interval remained below the same tolerance.

This combined criterion ensures that the monitored variable is not only statistically stable but also free from systematic drift. As a result, convergence is assessed in a robust and objective manner, accounting for both random fluctuations and trends in the CFD time series. As described in the simulation setup, an initial pressure of $P_0 = 7900$ Pa is applied, corresponding to a temperature of $T = 298.15$ K. Following the examples of Scandurra et al. [69], a tolerance of approximately 0.1% of the mean pressure is adopted. Therefore, the tolerance used in this case is

$$\Delta P = 0.001 \cdot P_0 \approx 8 \text{ Pa}. \tag{3.20}$$

For experiment III, meshes of 100,000, 300,000, 500,000, and 600,000 cells have been evaluated. As already mentioned in section 3.1.1, the considered meshes fulfilled common mesh quality requirements. As seen in Fig. 3.3, all 4 simulations generally show similar initial behavior while approaching a value slightly above 7900 Pascals as static pressure. The simulation employing a mesh resolution of 300,000 cells reaches convergence after a comparatively small number of iterations. A more detailed assessment using the previously described convergence criterion, based on the temporal evolution of the monitored quantity, allows identification of the time at which a steady state is achieved. As illustrated in the Appendix A.5, the solution can be considered converged after approximately 2,500 iterations. Furthermore, a comparison between the 300,000-cell mesh and the highest-resolution mesh of 600,000 cells reveals that both simulations attain nearly identical mean values in the steady state regime. The relative deviation between these two cases is on the order of $\approx 0.004\%$. Interestingly, the 500,000-cell mesh exhibited a slight upward outlier in static pressure. However, the convergence of the 300,000 and 600,000 resolutions suggests that spatial discretization errors are minimized, and the deviation at 500,000 cells might stem from sensitivity to specific grid topologies at that intermediate density. For these reasons, namely the rapid convergence behavior and the negligible difference in steady-state mean values compared to the finest mesh, the grid consisting of 300,000 cells is selected as the reference mesh for the subsequent heat transfer simulations conducted for the configuration of experiment III. This mesh corresponds to a physical average cell volume of

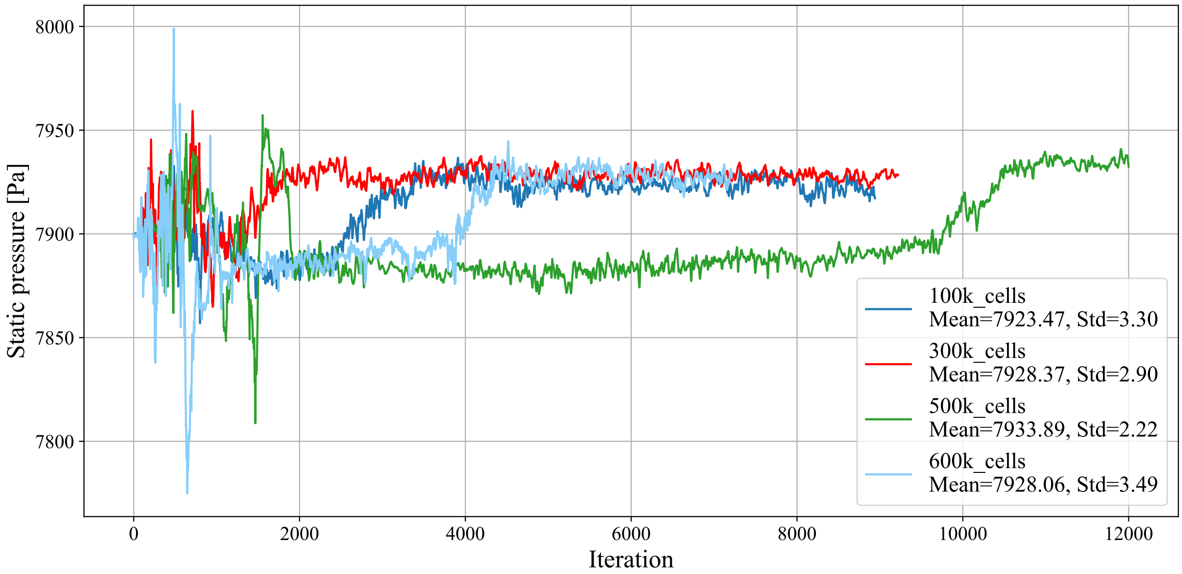

Figure 3.3: Grid independence study for experiment III: Convergence of the static pressure toward a steady-state value for different mesh resolutions

$$V_{\text{cell,avg}} \approx 5.59 \times 10^{-3}\,\text{mm}^3, \tag{3.21}$$

and an average characteristic cell length of

$$l_{\text{avg}} \approx 0.18\,\text{mm}. \tag{3.22}$$

The grid independence study for experiment II was performed following the same methodology as described previously. Mesh resolutions of 160,000, 320,000, 500,000, 700,000, and

1,000,000 cells were systematically investigated in order to evaluate the influence of spatial discretization on the numerical solution. Using the same convergence assessment algorithm, the standard deviation of the monitored quantity was evaluated after the simulations had reached steady state for each mesh density. The results of this analysis are summarized in Fig. 3.4. Among the tested configurations, the mesh consisting of 700,000 cells demonstrated the most favorable balance between numerical stability and computational cost, while the 1,000,000-cell mesh yielded the overall lowest standard deviation, amounting to approximately

$$\sigma_{\text{min}} \approx 0.05\%. \tag{3.23}$$

In addition, the steady-state mean value obtained with the 700,000-cell mesh remained in very close agreement with that of the finest mesh. The relative deviation between these two cases was in the order of

$$\Delta_{\text{rel}} \approx 0.01\%. \tag{3.24}$$

Indicating that further refinement beyond 700,000 cells provides only marginal improvement in solution accuracy. As illustrated in the Appendix A.6, approximately 3,000

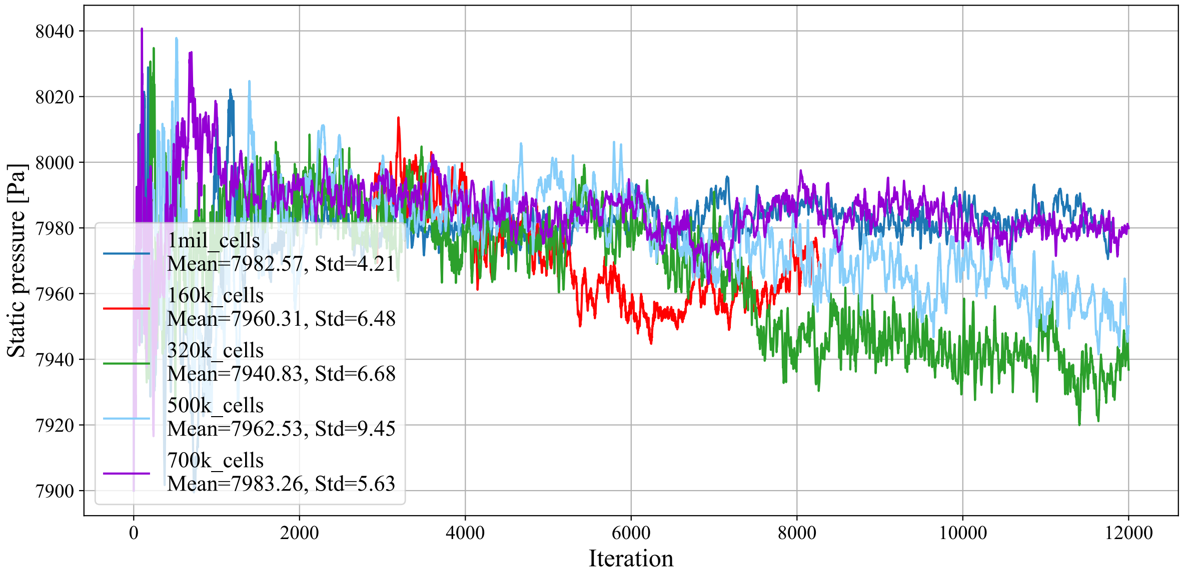

Figure 3.4: Grid independence study for experiment II: Convergence of the static pressure toward a steady-state value for different mesh resolutions

iterations were required for the monitored quantities to satisfy the prescribed convergence tolerance. For these reasons, the mesh consisting of 700,000 cells was selected for all subsequent simulations involving applied heat input. The chosen mesh results in an average cell volume of

$$V_{\text{cell,avg}} \approx 8.96 \times 10^{-3}\,\text{mm}^3, \tag{3.25}$$

and an average characteristic cell length of

$$l_{\text{avg}} \approx 0.21\,\text{mm}. \tag{3.26}$$

An overview of the mesh quality metrics of both considered meshes for the final simulations is provided in Tab. 3.1.

Table 3.1: Comparative mesh quality metrics for the numerical domains of experiment II and experiment III

Quality metric	Experiment II (700k cells)	Experiment III (300k cells)
Minimum orthogonal quality	0.640932	0.551
Average orthogonal quality	0.9444	0.94
Maximum skewness	0.4819	0.63
Average skewness	0.010434	0.016
Maximum aspect ratio	13	9

With the CFD methodology and numerical setup established, the focus now shifts to the approach using machine learning for predicting the thermal resistance of PHPs. The following chapter presents the corresponding machine learning framework developed in this thesis.

4. Machine Learning Model

Selecting an appropriate machine learning model is a crucial step in the data-driven modeling of engineering systems. In applications such as predicting complex heat transfer behavior, models must achieve high accuracy while working reliably with limited data and providing interpretable results. Variations in experimental conditions, data quality, and nonlinear relationships further increase the demands on the model.

The following section describes the model selection process and outlines why a particular algorithm was identified as the most suitable for the objective of this study.

4.1. Model Selection

The use of neural networks (NN) is a common and established approach for applying machine learning in industrial and research environments. The architecture is based on the operating principle of the brain. A collection of interconnected neurons (small processing units) communicates to solve specific problems. Artificial neural networks (ANNs) utilize these connections. Different connections have different weights, mimicking how synapses in the brain connect neurons with varying strengths. Different weights respond to specific stimuli in different ways, producing different outputs. As in the human brain, a large amount of data is required to make accurate predictions about the environment. Limited data leads to imprecise and unstable responses to stimuli. A model neuron, called a perceptron, "perceives" an input signal. It is processed through weights and a bias, producing an output value [64]. While neural networks are flexible and capable of effectively modeling nonlinear dependencies, they typically require large datasets, long training times, and extensive parameter tuning to achieve stable convergence. For smaller and structured datasets, this can lead to unstable or overfitted results. In a recent study conducted by S.H. Godasiaei et al. [26], machine learning methods were evaluated as an alternative to CFD simulations for heat transfer modeling, comparing ANNs to different ML algorithms such as random forest (RF) and the extreme gradient boosting (XGB) algorithm. The results show superior performance of the XGB algorithm in metrics like the mean absolute error MAE and RMSE (metrics introduced in section 4.2.3) compared to ANN and the RF model. The publication also suggests that, for a properly functioning neural network, the setup in the paper did not supply a sufficient amount of data for the NN to perform well. Furthermore, XGB allows the use of Shapley Additive ExPlanation (SHAP) analysis to evaluate outputs and interpret the trained machine learning model by quantifying the contribution of each input feature to the predicted output. ANNs are relatively opaque regarding how they process inputs. XGB also already has an open source implementation and documentation, CPU parallelization options, enabling efficient optimization with a manageable workload. It is computationally efficient, can handle datasets effectively, and provides competitive accuracy without the extensive overhead of deep neural networks

[37, 3]. Therefore, given the limited dataset size and the need for interpretability of the results, XGB was selected as the preferred model architecture. Based on these considerations, the following section presents a detailed description of the XGB algorithm, including its underlying principles, iterative learning strategy, and its applicability to modeling complex heat transfer phenomena in PHPs.

4.1.1. The XGBoost Algorithm

XGB, developed by Chen et al. in 2016, is an ensemble-based ML algorithm that builds sequential decision trees to minimize a specified loss function, offering high predictive accuracy, computational efficiency, and robustness [14]. XGB belongs to the family of gradient boosting algorithms, where decision trees are built iteratively to minimize the residual errors of the previous trees. The final model is an ensemble of smaller decision trees, often called weak learners, with each tree helping to correct the mistakes of the previous ones. This sequential process is known as tree boosting. The term "gradient" derives from the first derivative (gradient) and second derivative (Hessian) of the loss function that is utilized in its Taylor approximation, later introduced in Eq. 4.6. For a given data set D with n cases and m features $\mathcal{D} = \{(x_i, y_i)\}$ $(|\mathcal{D}| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$, a tree ensemble model uses K additive functions to predict the output as stated in Eq. 4.1 [14].

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}, \tag{4.1}$$

where $\mathcal{F} = \{f(x) = w_{q(x)}\}$ $(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$ is the space of regression trees. Here q represents an exemplary tree structure, with T representing the total number of leaf nodes and the corresponding leaf weights w. Thus, f_k is a function representing a specific tree structure that changes over tree iterations corresponding to the minimization of the loss function. A node is a decision point in the tree that splits data according to a criterion, whereas a leaf is a terminal point that assigns a constant prediction value to all instances within it. XGB does not natively support simultaneous optimization of multiple target variables. Multi-target prediction is therefore typically achieved by training independent trees or ensembles for each target, rather than optimizing all targets within a single tree or ensemble. As discussed earlier, neural networks improve by calculating the gradient of errors with respect to the weights. In XGB, the weights represent the values of the leaf nodes seen in Fig. 4.1. Each tree starts with a single node, where an initial threshold is set to cluster the initial residuals that arise from a first predicted output value for all cases. For each node, a quality score is calculated, essentially measuring how alike the clustered residuals numerically are:

$$Sc = \frac{1}{2} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda}. \tag{4.2}$$

The mathematical background of this score is explained later in this section. Under the mean squared error (MSE) loss function:

$$l = \tfrac{1}{2}(y_i - \hat{y})^2 \tag{4.3}$$

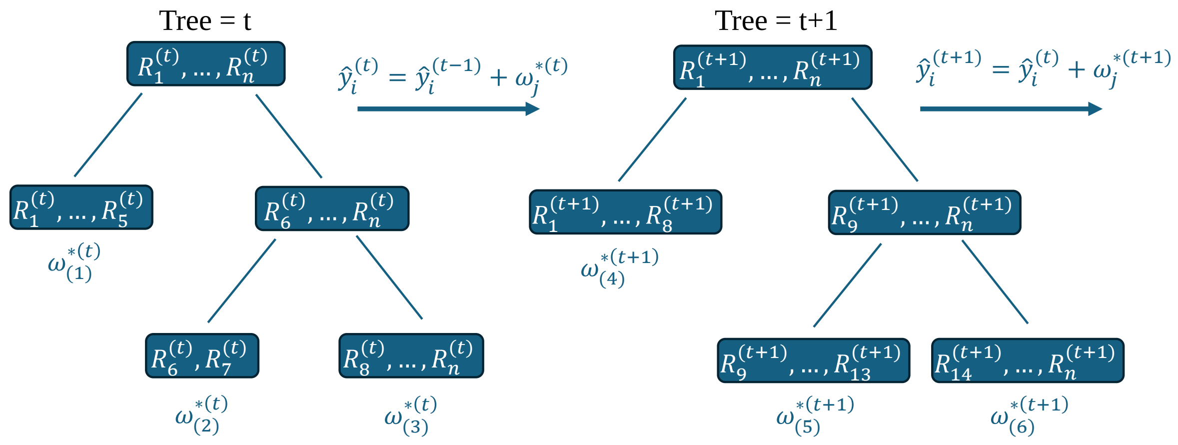

Figure 4.1: Structural representation of a gradient boosted decision tree ensemble

with y being the actual target value and $\hat{y}$ the predicted value, the score is defined as the ratio between the sum of residuals in the numerator and the number of residuals augmented by a regularization factor λ in the denominator. Accordingly, g_i and h_i denote the first and second derivatives of the loss function with respect to the predictions, i.e., the gradient and hessian, respectively. The first residual is commonly the difference between each output value of the dataset and the initially guessed value, i.e., the arithmetic mean of all predictions. The goal of the tree is to get really good at clustering residuals, or, in other words, categorize different cases in a way that similar outputs are sorted into the same leaf node of the tree, thus resulting in a similar prediction. For the next layer of nodes, the algorithm checks whether better predictions can be achieved by clustering similar residuals into new subgroups, i.e., choosing a new threshold to split them. Thresholds are evaluated iteratively for all possible features. For each candidate split, the algorithm calculates a value that is called the gain, introduced later on in Eq. 4.10, which is ultimately responsible for the decision if and where the features are split for residual clustering. According to the gain, the sum of the scores of the resulting nodes of the current split (left node and right node) is subtracted from the score of the parent node. This is done for all possible splits for all input features until the maximum gain is found. The maximum number of splits, corresponding to the tree depth, can be predefined or optimized through hyperparameter tuning to prevent overfitting. Hyperparameters are external configuration parameters that govern a model's learning behavior and structure. In XGB, several hyperparameters control the model's learning process and complexity. Common ones are, e.g., the learning rate and regularization parameters. The learning rate, denoted by η, scales the contribution of each newly added tree to the ensemble. By controlling how strongly each tree influences the overall prediction, the learning rate moderates the model's sensitivity to iterative updates, helping to stabilize training and reduce the risk of overfitting. Regularization, implemented via parameters such as λ and γ, penalizes overly complex trees to improve generalization. Specifically, λ applies regularization to the leaf weights, shrinking them toward zero. This effect is more pronounced in small leaves and diminishes in leaves containing many residuals. The parameter γ defines a minimum gain threshold for splitting. During tree construction, any branch that does not increase the objective function by at least γ is pruned. This pruning mechanism further limits tree complexity, preventing the addition of branches that contribute minimally to predictive accuracy. Further details on the hyperparameter optimization strategy and its implementation are presented in Section 4.2.3.

The predictions at each iteration are given by the leaf weights w, which are obtained by minimizing the objective function $L^{(t)}$. As in regular gradient boosting, prediction errors are quantified using a loss function $l(y_i,\hat{y}_i)$. The objective function is defined as:

$$L^{(t)} = \sum_{i=1}^{n} l\Big(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\Big) + \Omega(f_t) \tag{4.4}$$

Here, y_i corresponds to the observed value from the data set, $\hat{y}_i^{(t-1)}$ is the prediction at the previous iteration $t-1$, and the current tree's output value is $f_t(x_i)$. The loss function l measures the error between the new prediction $\hat{y}_i^t = \hat{y}_i^{(t-1)} + f_t(x_i)$ and the observed value (e.g., $l = \frac{1}{2}(y_i - \hat{y}_i^t)^2$). The regularization term:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{4.5}$$

discourages overly complex trees by penalizing the number of leaves T and shrinks leaf weights w, thereby reducing overfitting. With γ being a penalty encouraging pruning and T the number of leaf nodes in the tree. The goal is to minimize this loss function. The term for f_t can be approximated using a second-order Taylor approximation:

$$L^{(t)} \approx \sum_{i=1}^{n} \Big[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \Big] + \Omega(f_t) \tag{4.6}$$

where g_i and h_i describe the first- and second-order derivatives of the loss function for the current tree iteration t:

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), \quad h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}). \tag{4.7}$$

Since $l(y_i, \hat{y}_i^{(t-1)})$ does not affect the optimal output value for the current tree iteration, it is omitted. Setting $\gamma = 0$ during this derivation simplifies $\Omega(f_t)$ to a scaled version of f_t. By introducing j as the instance of a leaf and differentiating the simplified loss function with respect to the output value f_t and setting the resulting expression to zero, this yields the extremum for a specific node:

$$0 = \sum_{i \in I_j} g_i + \left(\sum_{i \in I_j} h_i + \lambda \right) f_t(x_i)\ . \tag{4.8}$$

By definition, $f_k(x_i) = w_{q(x_i)}$, meaning that w is the vector of leaf weights for all points x_i that fall into that leaf. Each column would represent a separate optimization problem for a leaf, so they are solved independently. Rearranging the equation for the leaf weights yields the previously introduced output value, which corresponds to the prediction assigned to each leaf in a given tree:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \tag{4.9}$$

Using Eq. 4.8 and replacing $f_k(x_i)$ with the derived optimal output, followed by multiplication by -1, yields the inverted loss function previously introduced. The maximum of the function is the quality score that determines the threshold set if a node includes more than 2 residuals and can still be split. In the implementation of the algorithm, it is actually two times that number, but it is insubstantial as long as every score is scaled the same. By solving for the maximum of the inverted form of Eq. 4.8, Eq. 4.2 is obtained. The formula for the gain, as introduced previously, is given by:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{4.10}$$

where G and H denote the sums of the first and second derivatives of the loss function (g_i and h_i) for each leaf.

While the previous discussion focused on regression with continuous outputs, which is also the task in this work, XGB operates similarly when applied to classification problems, where the target variable is categorical. Here, thresholds are still determined based on the input variable, but the outputs are categorized into classes instead of numerical values. For example, in analyzing drug dosages and their effectiveness, regression trees evaluate how effective a certain dosage is on a continuous scale (e.g., 1 to 10, where 10 indicates 100% effectiveness), whereas classification assigns discrete labels such as "effective" or "ineffective." Thus, classification relies on probabilistic predictions, although the underlying principle and formulas of regression and classification are closely related [51].

4.2. XGBoost Model for Predicting Pulsating Heat Pipe Performance

4.2.1. Feature Engineering and Selection

To acquire training data for the ML model, a large amount of published PHP data was collected, categorized, and interpreted. A final subset of 3,473 cases was retained for the final model, deliberately spanning operational heat generation ranges from LT- to HT-PEMFCs. Thermal resistances that lie outside an expected range and fill ratios below 5 % are neglected. The final ranges selected are shown in Tab. 4.1. Taking into account variances in experimental setups, a core assumption across examined results is an equilibrium of heat fluxes between the heat source and heat sink. Furthermore, the heat sinks

Table 4.1: Operational envelope and data constraints for the PHP dataset

Parameter	Lower bound	Upper bound
Thermal resistance	0.1 K/W	5 K/W
Heat input	1 W	250 W
Fill ratio	5%	100%

are considered ideal, implying that the manner in which the cooling is conducted has no impact on the thermal resistance. The compiled dataset was subsequently processed to define a consistent feature space for model training. After comprehensive feature engineering and reduction, the final set of parameters used for model training is summarized in Tab. 4.2. An essential step in feature preprocessing is avoiding strong multicollinearity among input features, because it is known to negatively affect model performance, often resulting in poor generalization and overly optimistic performance estimates [86]. In addition, features that are directly and physically correlated with the target output variable must not be included as inputs, as this would introduce data leakage, describing the phenomenon of information about the target being already known to the model in training. Geometric parameters were introduced to reduce the overall number of features, such as the hydraulic diameter, which enables a unified representation of different tube geometries and allows comparison with PHP concepts for bipolar plates as proposed by Franke et al. [24]. To obtain the heat flux per unit area, the input power was divided by the calculated evaporator area. The heat transfer area ratio is defined as the non-adiabatic outer pipe area (i.e., the effective area) divided by the total area of the PHP. In addition to the dimensionless parameters, a dimensionless geometrical parameter was formulated to characterize the pipe packing density within the PHP structure.

Table 4.2: Input feature space and target variable for the XGB model

Category	Type	Feature
Input	Original	Number of meanders
Input	Original	Area ratio of condenser/evaporator
Input	Original	Fill ratio (FR)
Input	Original	Inclination
Input	Original	Hydraulic diameter
Input	Original	Heat flux through evaporator
Input	Original	PHP length
Input	Original	Ratio of heat transfer area
Input	Dimensionless	Jakob number
Input	Dimensionless	Kutateladze number
Input	Dimensionless	Bond number
Input	Dimensionless	Ohnesorge number
Input	Dimensionless	Pipe areal density
Output	Target	Thermal resistance

For the dimensionless numbers, a fluid data dictionary was built with data on commonly used working fluids for PHPs at the saturation level. The data was mostly based on the National Institute of Standards and Technology (NIST) database or publications on measurements of e.g., self-rewetting fluids [52, 1]. In the context of PHPs the performance underlies some fundamental physical principles. The key mechanisms by which heat is transferred in a PHP are the absorption and rejection of sensible heat in a working fluid, as well as the latent heat that is released during the phase change of a fluid. A dimensionless number representing the relationship of these two mechanisms is the Jakob number:

$$\mathrm{Ja} = \frac{c_p \left(T_\infty - T_{\mathrm{sat}}\right)}{h_{lv}} \ . \tag{4.11}$$

It is defined by the ratio of sensible to latent heat of a fluid. For T_∞, the evaporator temperature is commonly used as a reference. T_{sat} is the saturation temperature of the fluid, which coincides with the temperature from which the corresponding fluid properties are taken. With the average evaporator temperature remaining unknown to the model, an assumed temperature difference $(T_\infty - T_{\mathrm{sat}})$ of 20 K is used to calculate the Jakob number. PHPs require a minimum heat input to operate effectively. However, when the applied heat load exceeds the maximum heat that the working fluid can absorb prior to complete evaporation, dryout arises. Correlations based on the Kutateladze number are used for indicating this phenomenon, hence motivating the utilization of this number in this model [46].

$$\mathrm{Ku} = \frac{q''}{h_{lv}\, \rho_v \left(\frac{\sigma g (\rho_l - \rho_v)}{\rho_v^2}\right)^{1/4}} \tag{4.12}$$

The Ku-number relates the critical heat flux at the evaporator and the critical heat flux necessary for pool boiling of the fluid. With ρ_l and ρ_v being the densities of the liquid and vapor phase, respectively, σ the surface tension, and g the ruling gravitational constant. As an expansion of these dimensionless numbers, the Ohnesorge number was introduced, acting as a measure for the damping of oscillations induced by capillary forces [74]:

$$\mathrm{Oh} = \nu_l \sqrt{\frac{\rho_l}{\sigma D_h}} \tag{4.13}$$

with ν_l describing the kinematic viscosity of the liquid phase, and the hydraulic diameter D_h. The Bond number characterizes the critical tube diameter at which capillary forces dominate over gravitational forces. For proper PHP operation, capillary forces must prevail over gravitational forces as determined by the aforementioned Bond number given in Eq. 1.11.

4.2.2. Tree Construction of XGBoost

To gain a clearer understanding of the model's operation, a simplified example of the decision tree construction process in XGB is presented in the Appendix C. It illustrates the incremental boosting process of XGB, where each subsequent tree corrects the residuals left by the previous trees, improving prediction accuracy step by step. In practical applications, to ensure proper generalization and avoid overfitting, tree building and growth are controlled by using optimizable hyperparameters such as tree-depth limits, pruning strategies, and learning rates, limiting the change in prediction for each tree iteration. The procedure of hyperparameter optimization is explained in the preceding section.

4.2.3. Hyperparameter Selection and Optimization

To assess the performance of the model and enable characterization with parameters that lead to optimal results, the prediction error metrics used and referenced in the following sections are introduced in this overview:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{4.14}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4.15}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{4.16}$$

$$\mathrm{RMSPE} = 100\sqrt{\frac{1}{n}\sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{y_i}\right)^2} \tag{4.17}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}. \tag{4.18}$$

The mean absolute error (MAE) measures the average absolute error and is rather robust against large deviations, whereas the root mean squared error (RMSE) is more sensitive due to the quadratic penalization. The root mean squared percentage error (RMSPE) is a basic measure of the mean squared percentage deviation. The mean absolute percentage error (MAPE) measures the average absolute percentage deviation and is easier to interpret, as errors are not squared. It is used later for an extension of the model evaluation. The last error metric used is R^2, which essentially compares the prediction with the mean as a baseline prediction. A value of 1 describes a perfect prediction, while 0 means it is not better than simply taking the mean output value as a prediction for every case. The use of multiple error metrics is motivated by the fact that each metric captures different aspects of model performance. While MAE and MAPE provide an intuitive measure of average deviation, RMSE and RMSPE emphasize larger errors due to their quadratic formulation and are therefore more sensitive to outliers. In addition, R^2 evaluates the model performance relative to a simple mean-based baseline, providing a measure of explained variance. By combining these complementary metrics, a more comprehensive and transparent assessment of the model behavior is achieved, reducing the risk of misleading conclusions that could arise from relying on a single evaluation criterion.

Hyperparameters govern the learning behavior of a model and directly affect key aspects such as computational cost, accuracy, complexity, and generalization ability. Searching for the optimal set of hyperparameters can be formally understood as a high-dimensional optimization problem, as multiple parameters interact and the objective function, typically validation performance, is unknown and expensive to evaluate. Traditional approaches, such as random search (RS) and grid search (GS), are relatively uninformed. RS selects hyperparameter values uniformly at random within predefined bounds. While this method can eventually find good configurations, it is inefficient because many combinations must be evaluated before reaching an optimum. GS, on the other hand, discretizes the hyperparameter space into a structured grid and evaluates every possible combination. Although more systematic than RS, GS quickly becomes computationally prohibitive in high-dimensional spaces due to the exponential growth of combinations. Bayesian optimization (BO) provides a more intelligent alternative. By building a probabilistic surrogate model of the objective function, BO predicts which hyperparameter configurations are likely to improve performance and explicitly balances exploration of uncertain regions with exploitation of promising areas. This allows the algorithm to find optimal hyperparameters with fewer evaluations, leading to improved model performance in practice [72]. Generally, the problem of minimizing an unknown objective function is considered f that depends on a vector of input parameters $x \in X$, where X denotes the space of possible hyperparameter configurations. Thus, the combination of hyperparameters that leads to the minimization of a scoring metric, resembling f, such as the RMSE, must be identified. Since direct evaluation of f is expensive, a surrogate model is introduced. This surrogate represents a probabilistic belief over the behavior of the objective function based on the observations collected so far. The belief is updated iteratively as new function evaluations become available. An acquisition function is defined on top of the surrogate model. It quantifies the utility of evaluating the objective function at a candidate point in the parameter space by combining the predicted performance and the associated uncertainty. At each iteration, the acquisition function is optimized to select the next parameter configuration. The objective function is then evaluated at this point, and the resulting observation is incorporated into the surrogate model. This update refines the uncertainty estimates and the predictive distribution of the surrogate. This iterative process continues until a stopping criterion is met, gradually guiding the search toward regions of the parameter space that are likely to contain the minimum of the objective function.

In BO, a Gaussian process (GP) is used to model the unknown objective function by predicting both a mean and an uncertainty for each point in the hyperparameter space. In the used implementation, the default choice is a Matern kernel, which measures similarity between points and controls the smoothness of the function, together with automatic relevance determination (ARD) that assigns different importance to each input dimension. An acquisition function is used to balance exploration and exploitation, guiding the search efficiently [66].

Alternatively, the tree-structured Parzen estimator (TPE) replaces the Gaussian model with non-parametric densities that separately model "good" and "bad" hyperparameter configurations based on past observations. The next evaluation is chosen where the ratio of these densities is maximal. Both methods are considered Bayesian in nature [10]. For further information on the implementation and characteristics of both methods, refer to the official website of OPTUNA [57] and the implementation of the GP and TPE algorithm in the corresponding GitHub repository. [58, 59]. A direct comparison of TPE and GP within the same experimental setup shows that, although TPE requires less than half the computational time for hyperparameter optimization with $n = 200$ iterations compared to GP, GP consistently outperforms TPE across all used evaluation metrics. Hence, GP was chosen for the selection of hyperparameters in this study. The final ranges of hyperparameters selected for optimization are shown in Tab. 4.3. The lower and upper bounds were iteratively tailored depending on the performance and exploitation of BO. The RMSE was chosen as the evaluation metric as it is the standard for regression

Table 4.3: Defined search space and optimization constraints for the Bayesian optimization (BO)

Parameter	lower bound	upper bound
Number of trees	100	2500
Maximum tree depth	3	14
Learning rate	0.005	0.3
Subsample of cases	0.5	0.9
Subsample of input features	0.5	0.9
λ_{reg}	1e-5	750
α_{reg}	1e-5	100
Min. number of instances within leaf node	1	30
γ	0	10
Tree method = 'hist'; 'approx'	-	-
Evaluation metric = RMSE	-	-

tasks. The number of trees controls the number of boosting iterations, i.e., the number of decision trees sequentially added to the ensemble in the gradient boosting process. The learning rate is a scaling factor applied to each iteration of model predictions, regulating the model's sensitivity to predictions per iteration. Setting the subsample of cases to 0.5 means that, in each boosting iteration, only 50% of the available training instances are randomly selected to construct the next tree. This stochastic sampling strategy introduces additional randomness into the learning process, which helps reduce model variance and lowers the risk of overfitting. The sampling is performed independently at every boosting step.

The subsample of input features follows the same argumentative logic, but for the input features. For every tree in the boosting process, a new subset of columns is sampled, which increases model diversity and contributes to reducing overfitting. λ, γ are regularization parameters introduced in the previous section. Whereas α represents the L1-regularization term applied to the leaf weights in the boosting trees, encouraging sparsity and helping to prevent overfitting by effectively removing less important leaves from the model. For further information on all explained parameters and further parameters that can be tuned in theory, refer to the GitHub repository of the DML community [83].

To reduce the bias of a fixed training and testing dataset in final performance metrics of the model and to enhance the generalization performance, nested cross-validation (CV) is implemented [66, 40]. Nested CV consists of an outer loop of k iterations in which the entire dataset is split into k subsets, shown in Fig. 4.2 (k=5 in this example). Here, $k-1$ subsets are selected as training data and 1 set for the testing of the model, respectively. For each of the k iterations, a separate inner loop is initiated. This loop optimizes the hyperparameters based on a CV loop within the training dataset of the outer loop. When the inner loop is done, the hyperparameter combination that leads to the best model in this iteration is tested on the testing dataset from the outer loop at iteration k. The resulting metrics are compared for each of the k outer loop iterations. Finally, the model with optimal performance among all these iterations is selected. In the implementation of this work, both the inner and outer loops use $k = 5$ folds, resulting in five different

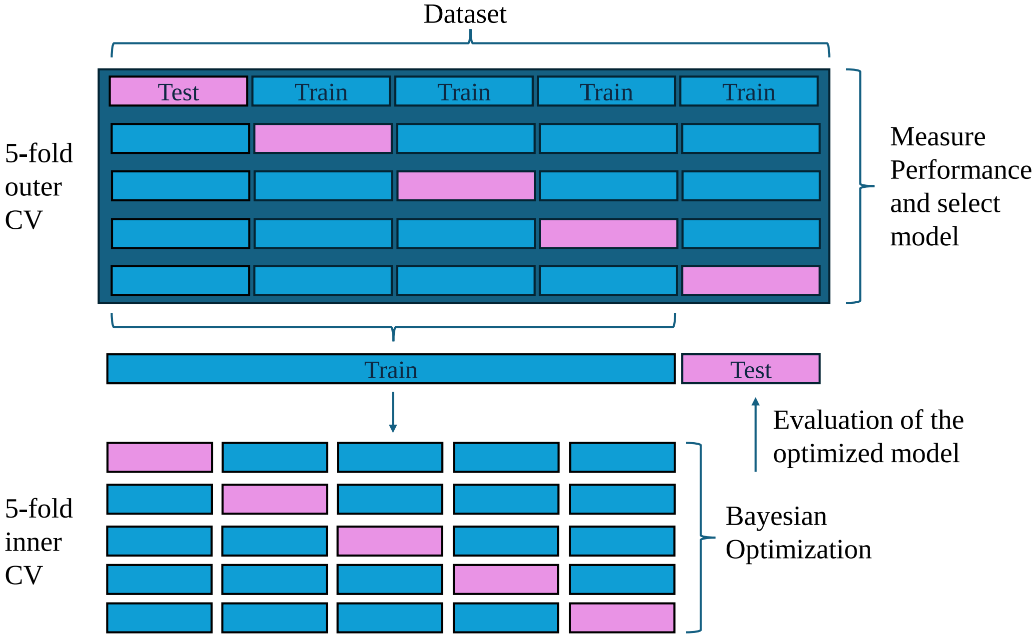

Figure 4.2: Schematic of the nested cross-validation procedure ($k = 5$)

training and validation datasets.

Given that a primary objective of this thesis is to provide a cost-effective and rapid approach for predicting PHP performance, the "Tree Method" parameter was analyzed in greater depth. This parameter decides how the possible number of feature splits is reduced to save computation time, but still achieve splits that lead to optimal model performance. The histogram-based method (hist) constructs trees using a histogram-based approximation, which discretizes continuous feature values into a fixed number of bins once before training. Each bin aggregates the gradient and Hessian statistics of the instances that fall into it during training, and candidate splits are evaluated only at bin boundaries rather than at every individual data point. Mathematically, if x_j is a feature with N samples, it is mapped into a discrete bin index b_k using quantiles such that each bin contains roughly N/B samples, where B is the number of bins. This reduces the computational complexity of split search from $O(N)$ to $O(B)$ while preserving predictive accuracy, making it efficient for large datasets [16]. The approximate greedy algorithm method (approx) also approximates candidate splits, but it does so using a weighted quantile sketch applied to the raw feature values. For further details of the algorithm, refer to Chen et al. [14]. The key difference between the two methods is how candidate splits are generated. Hist uses a greedy algorithm using histograms, optimized for speed, while approx constructs splits according to the Hessians of the features, which can be more precise in terms of capturing the weighted distribution but is generally slower and less memory-efficient than hist. Direct comparisons of the histogram-based method and the weighted quantile sketch in the same setup with 100 iterations for the BO per inner 5-fold and using the standard maximum number of discrete bins equal to 256, which defines the maximum number of possible split points per feature, revealed that the histogram-based method took only 52% of the time the weighted quantile sketch required for optimization. To ensure a fair and comparable evaluation of both methods, all relevant seeds were fixed to eliminate randomness from sampling and initialization. This means that any observed performance differences arise solely from the internal tree-building mechanisms. For a total of 5 repeated nested CV runs, the outer train/test splits were varied across iterations, but always using a fixed seed. This guarantees that both the histogram and approximate methods were evaluated on exactly the same training and testing splits in both the outer and inner CV loops. The mean differences in achieved performance metrics are summarized in Tab. 4.4. The histogram-based method reduced the optimization time and achieved negligibly better performance under identical data splits and optimization conditions. Therefore, the histogram-based method is selected for the final model.

Table 4.4: Comparative performance analysis between the 'approx' and 'hist' tree methods across 5 repeated nested CV iterations (relative mean difference in %)

Metric difference	Value
$RMSE_{\text{approx}} - RMSE_{\text{hist}}$	+0.19%
$MAE_{\text{approx}} - MAE_{\text{hist}}$	+0.26%
$RMSPE_{\text{approx}} - RMSPE_{\text{hist}}$	0.00%
$R^2_{\text{approx}} - R^2_{\text{hist}}$	0.00%

Further Model Refinements

The thermal resistance data is positively skewed, describing that lots of data and the median are on the left side of the mean, and a relatively small amount of data with great values on the right side, as seen in Fig. 4.3. For such distributions, a logarithmic transformation of the target variable is commonly used to improve modeling performance.

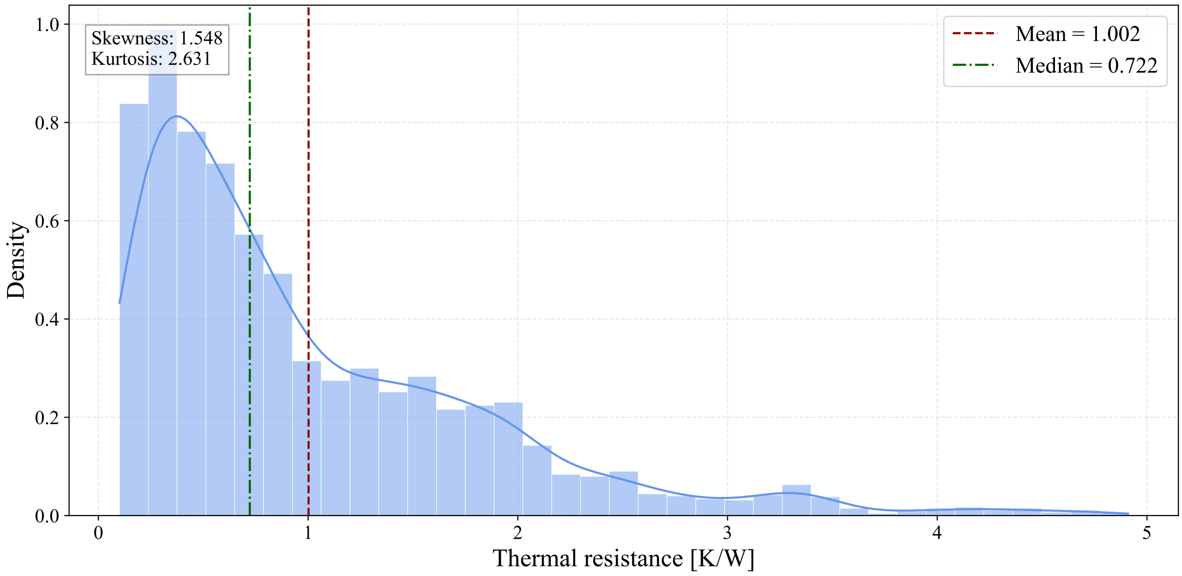

Figure 4.3: Distribution of the thermal resistance values

Following Manning et al.[47], a bias correction for either normally or non-normally distributed residual errors is proposed to account for the change in the objective function introduced by the logarithmic transformation. When the learning objective is MSE on logarithmically transformed targets:

$$\mathrm{MSE}_{\log} = \frac{1}{n}\sum_{i=1}^{n}\Big(\log(y_i) - \log(\hat{y}_i)\Big)^2 \tag{4.19}$$

the model is no longer minimizing the absolute error in the original space. Instead, for small errors, the log transformation implies:

$$\log(y_i) - \log(\hat{y}_i) \approx \frac{y_i - \hat{y}_i}{y_i} \tag{4.20}$$

so that the model minimizes for large relative errors.

Depending on how the residuals are distributed, a specific method of correction has to be selected. For the assumption of a normal residual distribution, Berkovits et al. suggest that skewness values above 3 and kurtosis values exceeding approximately 8 indicate severe non-normality for the distribution of data [11]. Skewness $\hat{\gamma}_1$ and kurtosis $\hat{\gamma}_2$ are calculated using the moment-based formulas [35, p. 76]:

$$\hat{\gamma}_1 = \frac{S^3}{(S^2)^{3/2}}, \quad S^3 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^3 \tag{4.21}$$

$$\hat{\gamma}_2 = \frac{S^4}{(S^2)^2} - 3, \quad S^4 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^4 \tag{4.22}$$

where $S^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2$ is the biased sample variance. These formulas correspond to the implementation used in Python's Scipy library [77]. The residuals of the model's prediction without correction are $\hat{\gamma}_1 = -0.15$ and $\hat{\gamma}_2 \approx 15$ for skewness and kurtosis, respectively, indicating non-normality. The distribution of the residuals is shown in Fig. 4.4. A quantile-quantile (Q-Q) plot, which compares the quantiles of the residuals to those of a normal distribution, shows some deviations in the tails due to a few extreme residuals for the testing and training datasets, indicating a high number of low residuals and some fairly high ones. In addition, a Shapiro-Wilk test was conducted, which essentially measures how well the residuals follow a normal distribution [71, 76]. The test revealed that the p-value is $<< 0.05$, rejecting the normality assumption. In addition to the nested CV procedure, an additional evaluation using a separate 70/30 train-test split was conducted. It should be noted that hyperparameter optimization was performed within the nested CV on the full dataset in order to identify a robust model configuration for subsequent design iterations under limited data availability. While the final model is therefore trained exclusively on the 70% training subset, such that the weights and split decisions of the model are estimated solely from this data, the chosen hyperparameter configuration reflects information from the structure of the entire dataset. Hence, the 30% testing set serves as an additional robustness check of predictive performance and supports the interpretation of improvements in predictive accuracy and overfitting behavior associated with the applied logarithmic transformation. To further assess generalization capability,

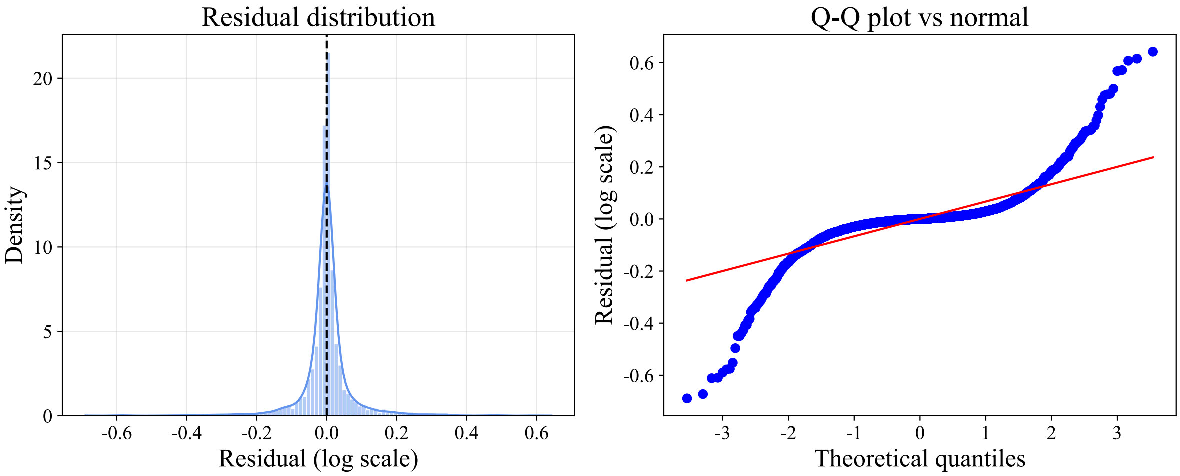

Figure 4.4: Analysis of residual distribution and normality for the complete PHP dataset on a logarithmic scale

the model is additionally evaluated on completely unseen external datasets in subsequent sections. Consequently, the nested CV results, presented in Sec. 5, remain the primary basis for performance assessment.

Regarding the distributional properties of the data, the skewness and kurtosis were calculated for both supplementary subsets. The training and testing datasets exhibit skewness values of $\hat{\gamma}_1 = -0.3$, and kurtosis values of $\hat{\gamma}_2 = 14$ and $\hat{\gamma}_2 = 18$. The corresponding plots illustrating these distributional characteristics are provided in Appendix A.9 and A.10.

In consequence of the results, Duan's smearing estimate is applied for the correction to account for the bias introduced by logarithmic transformation, under the assumption of homoscedasticity [18]. Homoscedasticity describes that the magnitude of the residuals is independent of the size of the predicted variable [82]. The corrected prediction on the original scale is given by:

$$\hat{E}[Y_0] = \exp(\hat{y}_0) \cdot \frac{1}{n} \sum_{i=1}^{n} \exp(\hat{\epsilon}_i), \quad S_m = \frac{1}{n} \sum_{i=1}^{n} \exp(\hat{\epsilon}_i) \tag{4.23}$$

where $\hat{y}_0$ is the prediction on the logarithmic scale and $\hat{\epsilon}_i$ are the corresponding residuals. The severity of possible heteroscedasticity is reduced because the optimization targets relative errors through the logarithmic transformation, resulting in residuals that are equally targeted independent of the target variable size, provided the data has consistent noise across both small and large values. However, in regions of higher thermal resistance where the PHP is still in the start-up phase and pulsation has not yet been established, the experiments may introduce additional noise and increased variance, thereby raising the likelihood of mild heteroscedasticity at higher thermal resistance values. To account for possible dependencies of the residual distribution on the magnitude of the thermal resistance, Duan's smearing estimate is applied in a stratified version as a similar idea of using smearing by group to account for heteroscedasticity between groups suggested by Manning et al. [47]. The residuals are grouped into bins based on quantiles of the predicted log-values $\hat{y}$ and a separate smearing factor is computed for each bin k:

$$\hat{E}[Y_0 \mid \hat{y}_0 \in \text{Bin } k] = \exp(\hat{y}_0) \cdot \frac{1}{n_k} \sum_{i \in k} \exp(\hat{\epsilon}_i) \tag{4.24}$$

where n_k is the number of observations in bin k. If the residuals are homoscedastic, this stratification changes nothing meaningful, as the mean of $\exp(\hat{\epsilon}_i)$ remains approximately constant across bins. However, in the presence of slight heteroscedasticity, binning prevents the correction from being disproportionately influenced by dominant tails in higher regions, thereby avoiding over-correction of smaller residuals. Importantly, the bin edges and smearing factors are computed exclusively on the training data. For new predictions and validation, the corresponding training bin is assigned, and the associated training smearing factor is applied. This approach avoids data leakage and prevents overly optimistic performance estimates. The average correction factor S_m over all bins is 1.0036, and the number of bins was chosen to yield approximately 23 residuals per bin as a trade-off between accuracy and noise reduction. The correction effect of 0.36%, with respect to the average thermal resistance of the dataset of $\bar{R}_{th} \approx 1$, is substantially smaller than the variability of the nested CV estimate of the MAE (5.7%, see Sec. 5), indicating that it lies within the statistical uncertainty of the performance estimation.

Careful feature engineering remains essential for achieving strong model performance. The heat flux feature is highly skewed as seen in Fig. A.7, spanning several orders of magnitude. In many modeling approaches, such pronounced skewness would motivate a logarithmic or similar transformation to stabilize variance and improve numerical behavior [23, 33]. However, for tree-based ensemble methods, a transformation of this kind is not strictly necessary. Decision trees construct splits based on the relative ordering of feature values rather than their absolute scale. A monotonic transformation, such as a logarithm, preserves this ordering and therefore does not change the structure of the split points in terms of which observations fall to the left or right of a threshold. But research done on this topic suggests slight performance improvements with logarithmic transformations even on input features for tree-based ensemble methods like XGB, hence the heat flux was put on a logarithmic scale before training (logarithmic distribution shown in Appendix A.8) and showed a tendency to improve the predictions in comparison to a model without the logarithmic transformation [68]. Therefore, the heat flux is used on a logarithmic scale.

The plot shown in the Appendix A.11 illustrates the predictive performance of the model over the iterations of decision trees. The model corresponds to the best-performing configuration identified through one iteration of nested CV and BO without logarithmic transformation of the target variable. Furthermore, the plot provides an overview of the considered error metrics for the final predictions of the model. For this setup, both RMSE and MAE are low for the training and testing data, with the predictions on the training set outperforming those on the testing set, indicating a tendency toward overfitting. The RMSE is an absolute error metric, whereas the RMSPE is a relative error metric expressed in percentage terms. Accordingly, the apparently higher RMSPE is not surprising, since the majority of target values are rather small, as shown in Fig. 4.3. In this regime, a small absolute RMSE can translate into a seemingly larger relative error.

After applying a logarithmic transformation to the thermal resistance and correcting for transformation-induced bias, the RMSPE, evaluated on the original scale, shows a substantial improvement, consistent with expectations. The corresponding plot is shown in the Appendix A.12. The model was built on the same train/test splits as the previous model, without logarithmic transformation of the thermal resistance, to ensure comparability. Because RMSPE emphasizes relative rather than absolute deviations, the transformation leads to a marked reduction in proportional prediction errors. This approach seemingly also improves the tendency to overfit by the previous models if the differences in error metrics of training and testing are compared. R^2 shows major improvements on the testing set, meaning the predictions outperform the mean by a great margin. While no metrics suggest a degradation of the fit on the training data, the application of the logarithmic transformation yields consistent improvements across all performance metrics on the testing dataset.

Model Interpretation with SHAP

In the case of studying PHP behavior, the interactions of the various features and their combined impact on the thermal performance are essential. To gain a comprehensive understanding of how the model evaluates the importance of different input features and how they interact in making predictions, a more detailed examination of each of the input features contributions to the model's prediction is necessary. In particular, it is of interest to identify which input features most strongly influence the model's predictions, the magnitude of their effect on the target variable, and whether this influence varies significantly across individual cases. Such an analysis enables the identification of model uncertainties, the detection of potentially irrelevant or misleading features, and the assessment of whether the learned relationships are consistent or strongly case-dependent. A Shapley Additive exPlanations (SHAP)-based feature importance analysis, based on the concept introduced by L. S. Shapley, was conducted for this purpose. It essentially measures the contribution of each input feature to the model predictions of thermal resistance. A more detailed explanation of the mathematical background of Shapley values, as well as an exemplary calculation, can be found in Appendix D [43].

5. Results

Following the theoretical formulation and the description of the Shapley value computation, this section presents the results obtained from the developed machine learning model and CFD simulations. The objective is to apply the previously introduced concepts in practice and to gain deeper insight into the model's decision-making process. First, the predictive performance of the model is reported. Subsequently, the corresponding Shapley values are analyzed in order to quantify the contribution of individual features to specific predictions. This allows for a systematic interpretation of both global patterns and discrete feature importance.

The final XGB model was optimized using repeated nested CV on the full dataset. In the outer loop, a 5-fold CV was repeated five times with different random splits, resulting in 25 independent outer validation runs. For each of these 25 outer folds, the data was split into a training set (80%) and a validation set (20%). Within the training set of each outer fold, Bayesian hyperparameter optimization was performed using an inner 5-fold CV loop with 300 iterations. From the resulting 25 optimal hyperparameter configurations (one for each outer fold), the configuration that achieved the lowest RMSE on the corresponding outer validation set was selected as the final hyperparameter configuration. Overall, nested CV provides a robust estimate of the model's predictive performance, as the outer test folds remain completely unseen during training and hyperparameter tuning. The final hyperparameters selected through nested CV are presented in Tab. 5.1. A comparison of the performance metrics obtained by averaging the results of the 25 outer validation folds in the nested CV, thereby avoiding overly optimistic performance estimates, and those from the 30% test subset after training on the 70% subset is provided in Tab. 5.2. The 70/30 train-test split was introduced in Sec. 4.2.3. The training on the 70% subset was conducted using the optimized hyperparameters listed in Tab. 5.1. The

Table 5.1: Final hyperparameter configuration of the optimized XGBoost model following Bayesian optimization on the full dataset

Parameter	Value
Number of trees	2289
Maximum tree depth	11
Learning rate	0.1
Subsample of cases	0.9
Subsample of input features	0.8
λ_{reg}	239.3940
α_{reg}	0.0001320
Min. number of instances within leaf node	1
γ	0.0
Tree method	hist
Evaluation metric	RMSE

Table 5.2: Performance metrics from nested CV and from the final model trained on the 70% subset and evaluated on the 30% test subset

Metric	Nested CV (Mean ± SD)	Training data	Testing data
MAE	0.1115 ± 0.0064 K/W	0.0287 K/W	0.1077 K/W
RMSE	0.2018 ± 0.0143 K/W	0.0597 K/W	0.1936 K/W
RMSPE	$20.2\% \pm 2.1\%$	7.5%	20.5%
R^2	0.941 ± 0.008	0.995	0.941

table shows good agreement between the performance estimates obtained from nested CV and the predictive accuracy of the final model on the 30% test set, indicating consistent generalization behavior across both evaluation schemes. The $R^2 = 0.941$ of nested CV shows good accuracy, as it is close to the optimal value of 1. An average MAE of approximately 0.11 corresponds to an RMSPE of 20.2%. The results indicate good predictive performance and stable behavior, with performance comparable to NNs developed for predicting the thermal performance of smaller, less detailed PHP datasets [4, 62]. The observed difference between training and test performance is expected and remains within a reasonable range compared to earlier hyperparameter configurations that indicated increased overfitting, which is visually shown in the parity plot seen in Fig. 5.1. It displays the experimentally measured thermal resistance on the x-axis and the predicted thermal resistance on the y-axis and shows an agreement of the majority of predictions with the measured experimental data. The tendency of predictions being more accurate in lower regions of thermal resistance is consistent with the previous assumption of slight heteroskedasticity of the data due to increased noise occurrence in higher regions of thermal resistance, due to startup problems, and the fact that generally fewer data points exist for high thermal resistances.

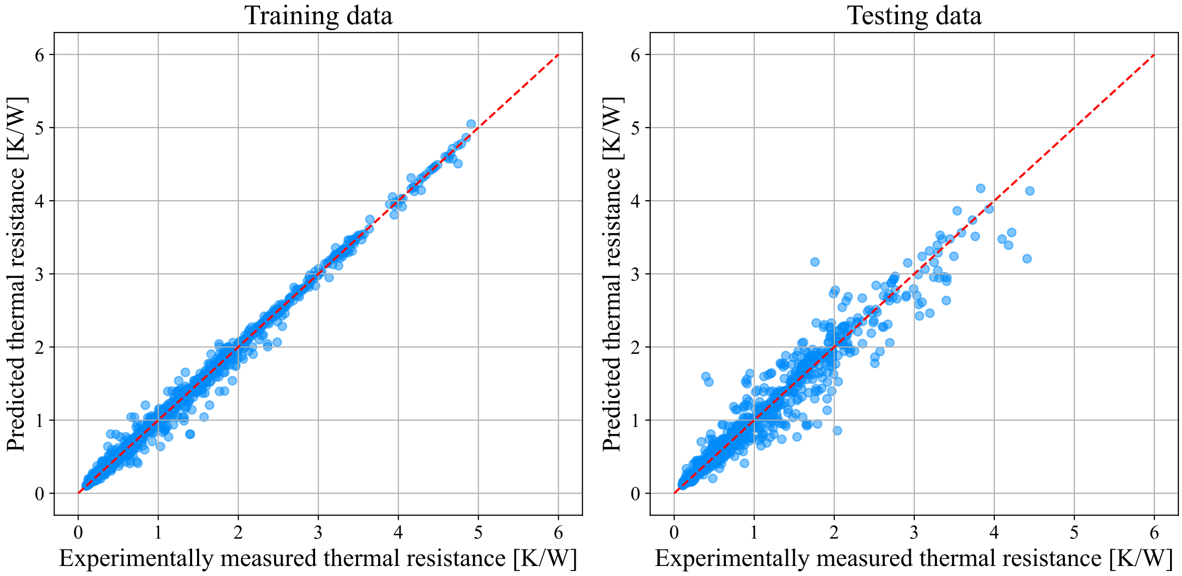

Figure 5.1: Parity plot of the final models' predictions

A beeswarm plot calculated through Shapley values, shown in Fig. 5.2, of the model fit onto the entire dataset, indicates a confirmation of that statement. In this plot,

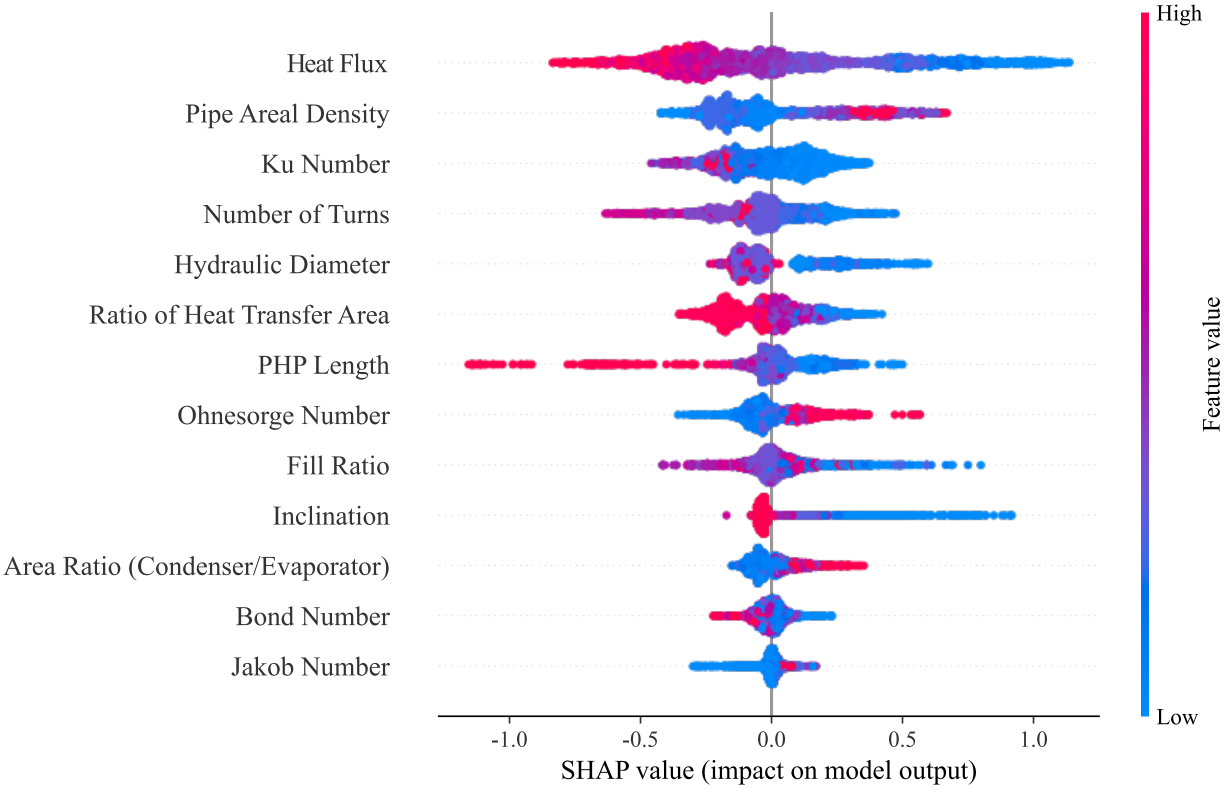

Figure 5.2: Beeswarm SHAP plot illustrating the impact of input features on the predicted thermal resistance

each input value gets its own range of values mapped onto the color scale individually, meaning the highest value receives the most vibrant color of red, and the lowest value receives the most vibrant color of blue. The x-axis displays the contributions of the features to the final thermal resistance predicted by the model. Here, each dot in each row of input features corresponds to one of the 3473 cases used in this fit. Furthermore, the higher the input is ranked from bottom to the top of the plot, the bigger its overall absolute impact on the predictions. To return to the previous point made, higher heat fluxes generally decrease the thermal resistance until a point of dryout occurs, where the majority of fluid evaporates, thus no effective heat transfer is possible. This point is mostly avoided by the majority of publications. But the startup is measured by most of them. So, focusing on the blue dots of the heat flux, it is visible that the density of the dots is lower compared to higher values of heat flux. As this is the region where PHP startup problems occur, and the feature contribution to the thermal resistance is positive, a higher uncertainty of predictions can be assumed for low heat inputs and thus higher thermal resistances, as stated before. The heat flux with the highest average absolute contribution to the model's predictions is followed by the pipe areal density. Notably, with increasing pipe areal density, the pulsating heat pipes tend to increase in thermal resistance. Higher density means more pipes on a smaller area, thus technically allowing more mass flow of the working fluid and thus more heat transport. The central parameter used to calculate the pipe areal density is the number of turns and the turn radius of the meander structure of the PHP. If increasing pipe areal density increases the thermal resistance, this might indicate that greater meander radii are favorable in designing PHP, which might be attributed to a lower resistance and local pressure drop in the meanders as suggested by [61]. As the Kutateladze (Ku) -number describes the ratio of the heat flux and the critical heat input required for evaporation, higher Ku-numbers suggest potentially improved heat transport until a certain point at which the system approaches the dryout. This statement is supported by the fact that the most negative contributions to thermal resistance are found in rather violet regions of the scale, and the red dots have the tendency to impact the thermal resistance less negatively than slightly lower Ku-numbers. An increasing diameter enables more latent heat transport because of higher mass flows under the constraint that the maximum Bond number is not exceeded. This would lead to inferior plug and slug formation, thus disabling the functionality of pulsating heat pipes. The sixth most impactful feature is the active area that participates in heat transfer. If there is more area that transfers heat, a more even distribution of heat and better heat transfer can be realized. This parameter alone does not give insight into how the active areas should be distributed. The area ratio of the condenser to the evaporator, as the fourth least important input feature for the model, indicates that a greater condenser area in relation to the evaporator area increases the thermal resistance. Hence, a higher evaporator area reduces thermal resistance for the model's predictions. An increasing number of turns decreases the thermal resistance, in consistency with expectations, that more fluid transport enables more heat transport from the evaporator to the condenser side. The parameter describing the intensity of how the fluid's viscosity mitigates the pulsation in comparison to the forces facilitating pulsation effects, so the inertia and surface tension of the fluid, is the Ohnesorge (Oh) number. As expected, the higher the viscosity in relation, the higher the Oh-number and the higher the thermal resistance. Another important parameter is the fill ratio (FR) of the PHP. If the PHP is supposed to operate in higher regions of temperatures, higher FRs are chosen to decrease the probability of dryout for a specific working fluid. It is evident that considering the FRs in the datasets span from 5 to 100%, around 50–60% seems to be the optimal FR of PHP performance. Studies on the inclination of PHPs showed that a vertical bottom head mode, describing the evaporator being placed at the bottom side of the PHP, is the configuration with superior performance in almost every case, which is supported by the SHAP-plot. Bond and Jakob numbers show the least impact on the model. This is intuitive because a certain Bond number has to be achieved for good PHP performance, and the majority of experiments operate the PHP in these regions. Fig. A.13 presents a bar plot summarizing the mean absolute feature impact on the model's prediction of the target variable. It shows that, considering an average thermal resistance of $R_{th} \approx 1$ across the dataset, the Jakob number contributes on average about 2.5%, while the Bond number contributes around 4%. These are the lowest average contributions to the predicted thermal resistance. However, they are still not negligible. The diagram also indicates that the contribution of heat flux is, on average, nearly twice as large as that of the second most important feature. The contributions of the remaining features increase in an approximately linear and consistent manner, with a slight jump observed between the ratio of effective heat transfer area and the pipe areal density.

Additionally, an uncertainty estimation was conducted using SHAP. For this purpose, the machine learning model was repeatedly trained on a randomly selected subset of the 70% training dataset with replacement. It was used to fit the model while maintaining the optimized tree structure from Tab. 5.1. The resulting model was then applied to the full 30% subset to compute SHAP values for all cases. Across the 50 iterations, a distribution of SHAP values was obtained for each feature, from which the variance and standard deviation were calculated. This procedure quantifies how sensitive the feature contributions are to slight variations in the training data. The results indicate that heat flux exhibits the highest absolute standard deviation of approximately 0.0022. However, when normalized by the average absolute SHAP value, see Fig. A.13, this corresponds to only about 0.6 to 0.7%, indicating a relatively low variation in its contribution. In contrast, inclination shows the smallest absolute deviation, with a relative deviation of approximately 0.7%. The largest relative deviation is observed for the Jakob number, reaching approximately 3.4%, followed by other thermophysical dimensionless numbers such as the Bond and Ohnesorge numbers. This increased relative variability may be attributed to the diversity of the dataset, which includes 47 different working fluids and mixtures. Consequently, thermophysical parameters, particularly those not directly linked to dominant features such as heat flux, tend to exhibit greater variability due to sparsity and larger gaps in the feature space. Despite these differences, the absolute deviations across all features remain small. Therefore, the model predictions can be considered robust and stable with respect to variations in the training subsets. The observed trend further suggests that improving the coverage of the feature space, rather than merely increasing the variability of the input features, could enhance model stability even further.

To complement the insights obtained from the analysis of the ML-models predictive capabilities, the corresponding CFD results are presented. The CFD settings specified in Section 3.1.2 are summarized in Appendix B.3 and Appendix B.4. The thermal resistance was determined based on the averaged surface temperatures of the evaporator and condenser sections, divided by the heat input and averaged over the oscillations. For the entirety of the simulations of experiment II, the simulation significantly underestimated the thermal resistance as seen in Fig. 5.4. The heat fluxes ranged from ≈ 3,600 W/m^2, corresponding to 10 W, up to ≈ 20,000 W/m^2, corresponding to a heat input of 55 W at the evaporator. For heat fluxes on the lower end, the startup time until oscillation of the thermal resistance was 24 seconds for 10 W and 18 seconds for 15 W. Apart from the dependence of the startup time on the heating power, increasing heat inputs also increases the oscillation frequency. However, as thermal resistance constitutes the central parameter of interest in this work, no further investigation of oscillatory characteristics was performed. In comparison, the specific heat fluxes at the evaporator for experiment III were significantly higher. In these simulations, the lowest heat input of 14.8 W already corresponded to a specific heat flux of 39,528 W/m^2. The first oscillations started after 4 seconds, as seen in Fig. 5.3, during which the thermal resistance was in good agreement with the experimental results. However, the evaporator temperatures exceeded those experimental measurements by a large margin, and the oscillations stopped after dryout occurrences in the evaporator sections after ≈ 10 seconds. After 50 seconds of physical simulation time, the evaporator temperature was still increasing beyond the experimental measurements, converging to a thermal resistance of ≈ 6, which is 60% higher than the experimental measurement. This deviation between simulation and experimental measurement of the thermal resistance increased further for the next higher simulated heat input of 32.1 W (Fig. A.15). Therefore, simulations with even higher heat fluxes were not conducted, as the discrepancy between the experiments and the simulations continued to

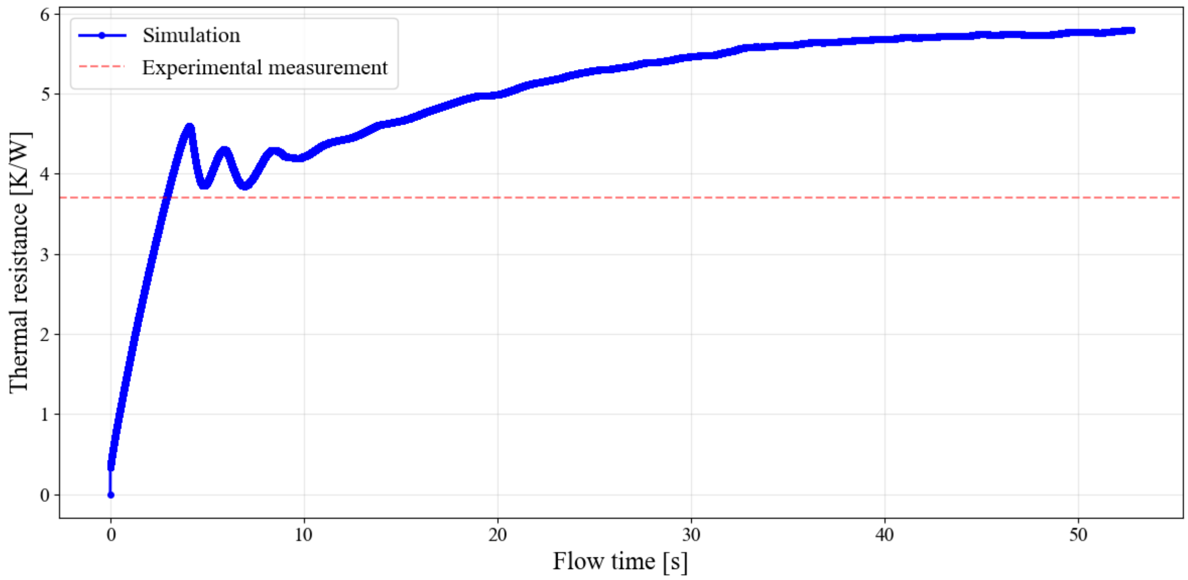

Figure 5.3: Simulated thermal resistance of experiment III (14.8 W) using ethanol and a fill ratio of 60%

increase. A potential source of inaccuracy lies in the use of constant coefficients in the Lee model. As shown by Tan et al. [75], fixed evaporation coefficients only yield accurate results within a limited range of heat fluxes. A dynamic correction of the evaporation coefficient based on the local temperature would be preferable. However, fine-tuning and validating such a function for the present pulsating heat pipe case with ethanol exceeds the resources and scope of this work. Another reason for the discrepancy between measurements and simulations might be the insufficient mesh resolution to capture the thin liquid films surrounding the vapor plugs. Moreover, the dynamic formation of menisci and liquid films was not modeled, since a fixed contact angle was imposed at the wall to limit computational cost and complexity. In contrast, Zhang et al. [87] describe a physically consistent model accounting for velocity-dependent apparent contact angles and oscillating film thickness, which are essential for accurately predicting the thermal resistance of PHPs. However, implementing such a physically consistent film model would require significantly finer meshes to resolve the thin liquid films in the order of 100 µm or less. Combined with the dynamic Lee Model coefficient correction and dynamic contact angles, this would lead to excessively high computational costs and further risks of numerical instabilities. The present simulations already suffer from very long simulation times, clearly demonstrating the enormous complexity involved in the accurate modeling of PHPs.

A direct comparison of the thermal resistances predicted by the ML model and simulated by the CFD of experiment II is shown in Fig. 5.4. The dataset used for training the ML model for these predictions excluded all samples corresponding to experiment II with an FR of 55%, ensuring that this case was not included in either training or hyperparameter optimization. This resulted in slightly different hyperparameters compared to those found through optimization on all available data (seen in Tab. 5.1). However, the difference in predictive performance was negligible. The RMSPE between the predictions and the mea-

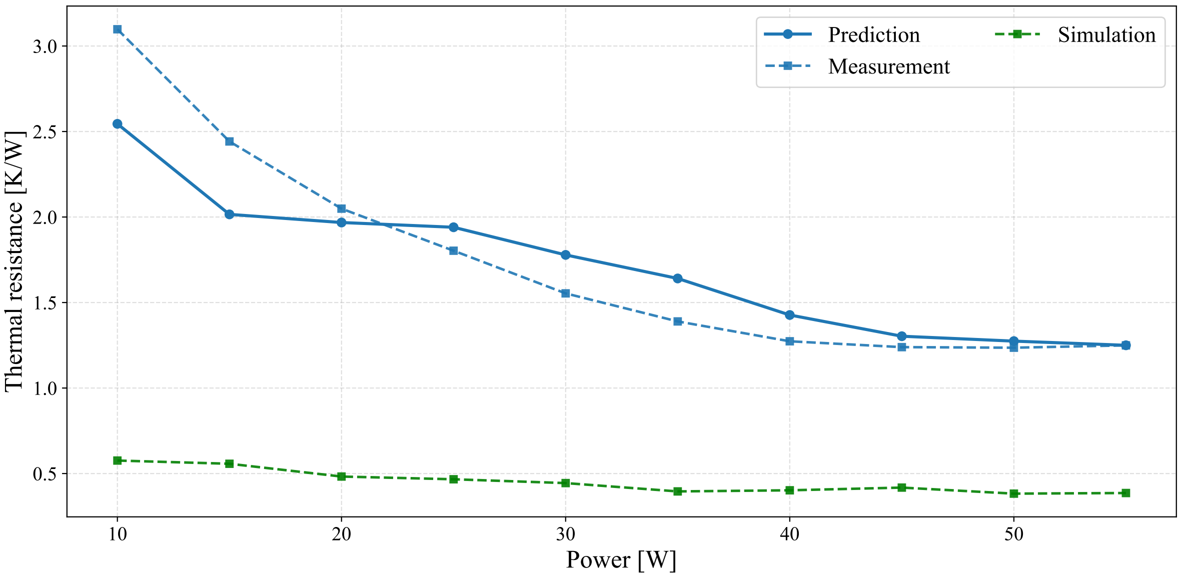

Figure 5.4: Predicted vs measured vs simulated thermal resistance of experiment II | RMSPE (pred. and meas.) = 11.899% | RMSPE (sim. and meas.) = 72.635% | MAPE (pred. and meas.) = 9.983% | MAPE (sim. and meas.) = 72.497%

surements is 11.899%, while the RMSPE between the simulation and the measurements is 72.635%. The RMSPE for the predictions is slightly pessimistic due to larger deviations at lower thermal resistance values. The average percentage deviation between the predictions and the experimental measurements is given by MAPE = 9.983%. The trends of decreasing thermal resistance with increasing power, as well as the magnitude of the thermal resistance, are predicted by the ML model with high accuracy. It outperforms the simulation in predictive accuracy, although the comparison may be unfair due to the lack of accuracy of the CFD model. It should also be noted that for the predictions seen in Fig. 5.4, the ML model was trained on data from experiment II with different working fluids and fill ratios, while this specific combination of working fluid and FR was not included in the dataset. Fig. A.14 summarizes the entire experimental results of the thermal resistance in comparison with the predictions of the ML model after excluding all data from experiment II in training and the final fit. It can be observed that the model is able to predict the trends present in the PHP, but it struggles to accurately capture the magnitude of the thermal resistance. With increasing heat input, however, the prediction accuracy improves. The comparatively high RMSPE of 47.1% for experiment II can be attributed to the fact that the measured thermal resistance values are systematically higher than those of most other, similar geometries in the dataset, effectively making this case an outlier. As a result, the model tends to underestimate the thermal resistance. To support this claim, another run of predictions was conducted on experimental results by D. Bastakoti et al. B.1, which was not included in the dataset at any point during training for this demonstration. The predictions are displayed in Fig. 5.5. These predictions indicate a systematic bias in the model that correlates with the average thermal resistance of the training data if the thermal resistance of an entirely new geometry is predicted. This limitation could be addressed by using a larger and more diverse dataset that includes greater variation in operating conditions and a wider range of geometrical

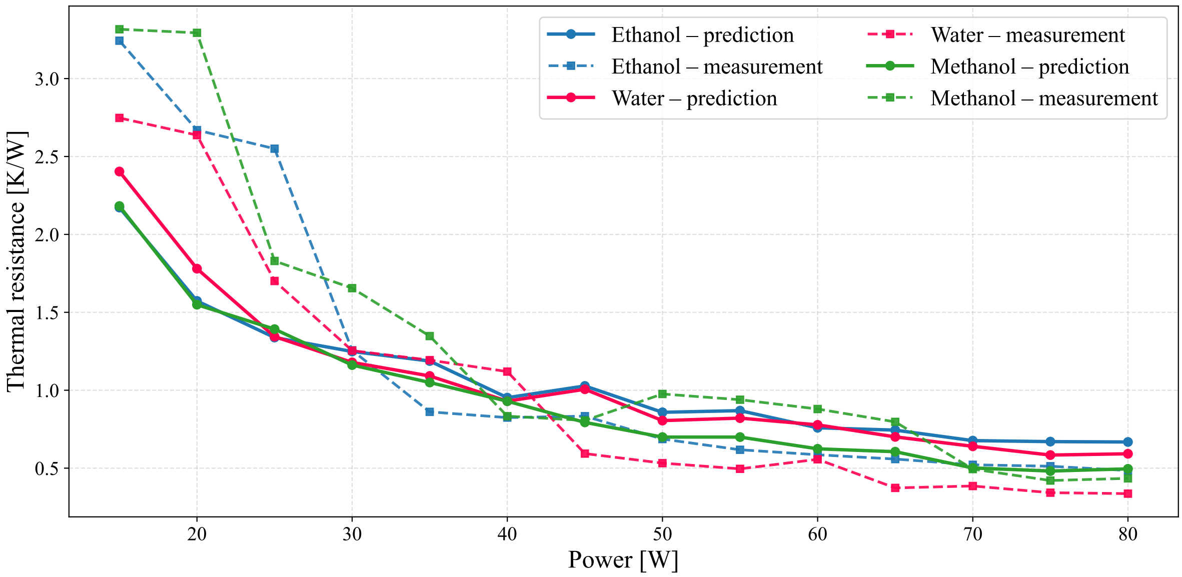

Figure 5.5: Predicted vs measured thermal resistance (Bastakoti et al. B.1) | RMSPE = 36.85% | MAPE = 32.44%

features, allowing the model to make better predictions on entirely new geometries. For the predictions shown in Fig. 5.5, the RMSPE is 36.85% and MAPE = 32.44%, showing improved predictive accuracy, compared to the predictions of experiment II. The trends of decreasing thermal resistance with increasing heat input are reasonably well captured by the predictions. The tendency of ethanol to perform slightly worse for this setup than methanol or water is also predicted for higher heat inputs and the corresponding heat fluxes.

In summary, the SHAP analysis provides insights into the machine learning decision process and the importance of features and their contribution to the final prediction, allowing a better understanding of the relative impact and relevance of the different input parameters. The machine learning model predicts thermal resistance with good accuracy. When predicting thermal performance for entirely new geometries, the model tends to revert towards the overall dataset mean while still capturing general trends, especially with increasing heat input. This behavior likely reflects the limited variability of geometric features in the dataset compared to more diverse input features computed from multiple other parameters and therefore span a much broader range of variability. The model performs particularly well on new cases with known geometries, while accuracy decreases for unseen geometries. Overall, the ML model outperforms the CFD simulations in the considered cases, which show larger deviations from the experimental measurements. Model performance is expected to improve further with more diverse datasets and a larger number of geometries.

6. Discussion and Outlook

Pulsating heat pipes (PHPs) offer new opportunities in thermal management systems for civil aviation. They could provide options for weight reduction due to their geometrical and operational flexibility. System complexity can also be reduced because of their operating principle, which does not require external control systems, cooling fluid reservoirs, or similar auxiliary components. When, for example, integrated into the bipolar plates (BPP) of fuel cells for civil aircraft, each cooling system of each BPP plate is separated, increasing redundancy and reducing the probability of system failure. Furthermore, this integration could lead to additional weight reduction when directly implemented into the existing spaces between BPP and MEA within the fuel cell system, which commonly serves as the cooling channels in which fluids are pumped through using external reservoirs.

Since the exact thermophysical phenomena of PHPs are complex, iterative development of new designs and performance prediction are essential for their effective design and optimization in future applications. Therefore, an approach using the XGBoost machine learning algorithm on existing experimental data is applied to obtain a comprehensive overview of the interactions between different design parameters and their influence on overall thermal performance, with thermal resistance serving as the defining thermal performance parameter, which is to be predicted by the model. This enables the prediction of new PHP designs by studying the relationships between PHP parameters. The model's predictions of validation data show satisfactory accuracy, with an average RMSPE of 20.2% and R^2 of 0.941, demonstrating the reliability of the model in predicting unseen data within the parameter range on which it was trained. The extrapolation capabilities of the model to unseen geometries, however, are limited. While general trends can be predicted with reasonable accuracy, the predictive accuracy of thermal resistance tends to decrease in precision if predictions are made on entirely new geometries. This limitation arises from the absence of a physical foundation within the model. Although relevant dimensionless parameters are included to account for thermophysical interactions, the model can only establish relationships within the parameter domains represented in the training data. Consequently, the model provides the most reliable and accurate results within the range covered by the training data, while predictions far beyond this range may show reduced accuracy.

As part of this work, CFD simulations were additionally performed to provide a physics-informed reference for comparison with the machine learning predictions and to support the interpretation of the observed results. There are several possible reasons for the discrepancies and inaccuracies observed in the CFD simulations. First, both the temperatures and heat fluxes were adapted from experimental data. A fixed condenser temperature was imposed based on thermocouple measurements, while the heat flux was assumed to be nearly ideal due to perfectly insulated conditions, as reported in the experimental setup. Another source of inaccuracy may be an insufficiently refined mesh. The grid independence study was performed under simplifying assumptions to reduce computational cost, as running full PHP simulations on significantly finer meshes was not feasible. As a result, the selected mesh may not adequately resolve the thin liquid films, even though these films play a dominant role in the heat transfer mechanism through latent heat transfer. In addition, inaccuracies may arise from the use of fixed Lee model coefficients in the VOF approach. A constant evaporation coefficient is likely only valid within a limited range of heat fluxes and temperatures. This limitation may be reflected in the results, where the heat fluxes of the simulated experiments II and III conducted by S. Khandekar in 2004 show a better agreement with the experimental measurements within a range between the investigated heat flux levels.

In its current state, the machine learning model is limited to relatively strict geometrical configurations, specifically bottom heating mode and planar meander-type structures. The model's predictive capabilities and stability are expected to further improve with better coverage of the feature space. The introduction of the hydraulic diameter allows for the extension of future studies to different pipe cross sections. Additionally, the fluid property database can be expanded with minor adjustments to the calculation methods of dimensionless parameters, particularly for cryogenic coolants for which already several experiments have been conducted and published. The development of correlation factors between, for example, circular (the ones used in this study) PHPs and flat plate PHPs could significantly broaden the available dataset, thereby improving generalization capabilities and supporting the interpretation and design of future PHP configurations for cooling systems in aerospace propulsion systems and applications.

The CFD model can further contribute to understanding the pulsating behavior of PHPs and the underlying physics. However, in agreement with previous publications, the results of this work suggest that CFD simulations of entire PHP systems remain of limited reliability under the employed simplifications. Moreover, they are computationally expensive when finer meshes, more complex solvers, or stricter convergence criteria are used, making them impractical for analyses and simulations of entire PHP systems.

To conclude, the motivation of this work was to develop an approach for predicting PHP performance and to facilitate the interpretation of their behavior in an efficient and sufficiently accurate manner. The machine learning results show a solid predictive capability and, in combination with SHAP analysis, provide improved insight into the relationships between input features and the predicted thermal resistance. While the current model already shows good performance, further optimization and additional data can further improve its accuracy, increasing its value for future PHP design iterations and optimization tasks.

Bibliography

[1] 3M. 3m™ fluorinert™ electronic liquid fc-72, 2015. URL https://multimedia.3m.com/mws/media/648920/3m-fluorinert-electronic-liquid-fc-72.pdf.

[2] H. Aburema, B. C. Hanson, M. Fairweather, and M. Colombo. *A generalised multiphase modelling approach with heat transfer and thermal phase change.* International Journal of Heat and Fluid Flow, 2024. doi: 10.1016/j.ijheatfluidflow.2024.109524. URL https://www.sciencedirect.com/science/article/pii/S0142727X24002492?via%3Dihub.

[3] P. V. Acharya, M. Lokanathan, A. Ouroua, R. Hebner, and Strank, S. Bahadur, V. *Machine Learning-Based Predictions of Benefits of High Thermal Conductivity Encapsulation Materials for Power Electronics Packaging.* Journal of Electronic Packaging, 2021. doi: 10.1115/1.4052814. URL https://asmedigitalcollection.asme.org/electronicpackaging/article/143/4/041109/1122828/Machine-Learning-Based-Predictions-of-Benefits-of.

[4] M. H. Ahmadi, M. Sadeghzadeh, A. H. Raffiee, and K.-w. Chau. *Applying GMDH neural network to estimate the thermal resistance and thermal conductivity of pulsating heat pipes.* Engineering Applications of Computational Fluid Mechanics, 2019. doi: 10.1080/19942060.2019.1582109. URL https://www.tandfonline.com/doi/full/10.1080/19942060.2019.1582109.

[5] Akachi Hisateru. Structure of a heat pipe, 1990. URL https://www.lens.org/lens/patent/151-789-425-811-172/frontpage.

[6] ANSYS, Inc. *ANSYS Fluent Theory Guide – 14.7.5. Evaporation-Condensation Model.* 2024. URL https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v242/en/flu_th/flu_th_sec_mp_evap_cond.html.

[7] ANSYS Inc. *Meshing User's Guide.* 2025. URL https://ansyshelp.ansys.com/public/Views/Secured/corp/v251/en/pdf/ANSYS_Meshing_Users_Guide.pdf.

[8] ANSYS, Inc. *Ansys Fluent Theory Guide; Section 14.3: Volume of Fluid (VOF) Model.* 2025. URL https://ansyshelp.ansys.com/public/account/secured?returnurl=/Views/Secured/corp/v251/en/flu_th/flu_th_sec_multiphase_vof.html.

[9] ANSYS Inc. *Heat Transfer Theory: 5.2.1.5. Inclusion of the Viscous Dissipation Terms.* 2025. URL https://ansyshelp.ansys.com/public/account/secured?returnurl=//Views/Secured/corp/v251/en/flu_th/flu_th_sec_hxfer_theory.html%23x1-2210006.2.1.

[10] J. Bergstra, R. Bardemet, Y. Bengio, and B. Kégl. *Algorithms for Hyper-Parameter Optimization.* 25th Annual Conference on Neural Information Processing Systems (NIPS), 2011. URL https://www.researchgate.net/publication/216816964_Algorithms_for_Hyper-Parameter_Optimization.

[11] I. Berkovits and G. R. Hancock. *Bootstrap Resampling Approaches for Repeated Measure Designs: Relative Robustness to Sphericity and Normality Violations.* 2000. doi: 10.1177/00131640021970961. URL https://journals.sagepub.com/doi/10.1177/00131640021970961.

[12] S. Bhapkar, C. Sain, and S. Kazula. *Review and Evaluation of Hydrogen and Air Heat Exchangers for Fuel Cell-Powered Electric Aircraft Propulsion.* EASN, 2025. doi: 10.3390/engproc2025090062. URL https://www.mdpi.com/2673-4591/90/1/62.

[13] T. C. Cano, I. Castro, A. Rodriguez, D. G. Lamar, Y. F. Khalil, L. Albiol-Tendillo, and P. Kshirsagar. *Future of Electrical Aircraft Energy Power Systems: An Architecture Review.* IEEE Transactions on Transportation Electrification, 2021. doi: 10.1109/TTE.2021.3052106. URL https://www.researchgate.net/publication/348622318_Future_of_Electrical_Aircraft_Energy_Power_Systems_An_Architecture_Review.

[14] T. Chen and C. Guestrin. *XGBoost: A Scalable Tree Boosting System.* Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. doi: 10.1145/2939672.2939785. URL http://arxiv.org/pdf/1603.02754v3.

[15] T. C. de Goede, N. Laan, K. G. de Bruin, and D. Bonn. *Effect of Wetting on Drop Splashing of Newtonian Fluids and Blood.* Langmuir, 2018. doi: 10.1021/acs.langmuir.7b03355. URL https://pubs.acs.org/doi/10.1021/acs.langmuir.7b03355.

[16] DMLC XGBoost Contributors. *Tree Methods Documentation (treemethod.rst).* 2026. URL https://github.com/dmlc/xgboost/blob/master/doc/treemethod.rst.

[17] R. Dreiling, V. Dubois, S. Zimmermann, T. Nguyen-Xuan, P. Schreivogel, and F. Di Mare. *Numerical investigation of slug flow in pulsating heat pipes using an interface capturing approach.* International Journal of Heat and Mass Transfer, 2022. doi: 10.1016/j.ijheatmasstransfer.2022.123459. URL https://www.sciencedirect.com/science/article/pii/S0017931022009280.

[18] N. Duan. *Smearing Estimate: A Nonparametric Retransformation Method.* Journal of the American Statistical Association, 1983. doi: 10.2307/2288126. URL https://www.jstor.org/stable/2288126.

[19] ENEA. *ANSYS FLUENT 12.0 Theory Guide - 6.2.2 Mesh Quality.* 2009. URL https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/node167.htm.

[20] ENEA. *ANSYS FLUENT 12.0 User's Guide; Section 26.3.1: Choosing the Pressure-Velocity Coupling Method.* 2009. URL https://www.afs.enea.it/project/neptunius/docs/fluent/html/ug/main_pre.htm.

[21] A. Faghri and Y. Zhang. *Fundamentals of Multiphase Heat Transfer and Flow.* Springer eBooks Engineering. Springer, Cham, Switzerland, 1st ed. 2020 edition, 2020. ISBN 9783030221379. doi: 10.1007/978-3-030-22137-9. URL https://link.springer.com/book/10.1007/978-3-030-22137-9.

[22] S. Fang, C. Zhou, Y. Zhu, Z. Qian, and C. Wang. *Review on Research Progress*

of Pulsating Heat Pipes. Inventions, 2024. doi: 10.3390/inventions9040086. URL https://www.mdpi.com/2411-5134/9/4/86.

[23] C. Feng, H. Wang, N. Lu, T. Chen, H. He, Y. Lu, and X. M. Tu. *Log-transformation and its implications for data analysis*. Shanghai archives of psychiatry, 2014. doi: 10.3969/j.issn.1002-0829.2014.02.009. URL https://www.researchgate.net/publication/264500976_Log-transformation_and_its_implications_for_data_analysis.

[24] F. Franke, M. Kober, and S. Kazula. *Preliminary design and thermal evaluation of a hexagonal pulsating heat pipe for fuel cell applications*. Case Studies in Thermal Engineering, 2025. doi: 10.1016/j.csite.2025.107272. URL https://www.sciencedirect.com/science/article/pii/S2214157X25015321.

[25] A. C. Frey, D. Bosak, E. Madrid, J. Stonham, C. M. Sangan, and O. J. Pountney. *Thermal management in high temperature proton exchange membrane fuel cells for aircraft propulsion systems*. Progress in Aerospace Sciences, 2025. doi: 10.1016/j.paerosci.2024.101052. URL https://www.sciencedirect.com/science/article/pii/S0376042124000782.

[26] S. H. Godasiaei and H. A. Kamali. *Evaluating Machine Learning as an Alternative to CFD for Heat Transfer Modeling*. Microgravity Science and Technology, 2025. doi: 10.1007/s12217-025-10163-x. URL https://link.springer.com/article/10.1007/s12217-025-10163-x.

[27] J. Gu, M. Kawaji, and R. Futamata. *Effects of Gravity on the Performance of Pulsating Heat Pipes*. Journal of Thermophysics and Heat Transfer, 2004. doi: 10.2514/1.3067. URL https://www.researchgate.net/publication/245439574_Effects_of_Gravity_on_the_Performance_of_Pulsating_Heat_Pipes.

[28] N. Hasan, S. F. Anwer, and S. Sanghi. *Natural convection in a bottom heated horizontal cylinder*. Physics of Fluids, 2005. doi: 10.1063/1.1932311. URL https://www.researchgate.net/publication/236110596_Natural_convection_in_a_bottom_heated_cylinder.

[29] IEA - International Energy Agency. *Global Hydrogen Review*. 2023. URL https://www.iea.org/reports/global-hydrogen-review-2023.

[30] Intergovernmental Panel on Climate Change. *IPCC, 2021: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2023. doi: 10.1017/9781009157896.001. URL https://www.ipcc.ch/report/ar6/wg1/about/how-to-cite-this-report/.

[31] J. Jo, J. Kim, and S. J. Kim. *Experimental investigations of heat transfer mechanisms of a pulsating heat pipe*. Energy Conversion and Management, 2019. doi: 10.1016/j.enconman.2018.12.027. URL https://www.sciencedirect.com/science/article/pii/S019689041831361X.

[32] S. Kazula, S. de Graaf, and L. Enghardt. *Review of fuel cell technologies and evaluation of their potential and challenges for electrified propulsion systems in commercial aviation*. Journal of the Global Power and Propulsion Society, 2023. doi: 10.

33737/jgpps/158036. URL https://www.researchgate.net/publication/368397235_Review_of_fuel_cell_technologies_and_evaluation_of_their_potential_and_challenges_for_electrified_propulsion_systems_in_commercial_aviation.

[33] O. N. Keene. *The log transformation is special.* Statistics in medicine, 1995. doi: 10.1002/sim.4780140810. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780140810.

[34] S. Khandekar. *Thermo-hydrodynamics of Closed Loop Pulsating Heat Pipes.* Doctoral thesis, University of Stuttgart, Stuttgart, 2004. URL https://elib.uni-stuttgart.de/items/24c50ac2-6cdd-433e-a7f1-a9e7224320f0.

[35] R. B. Kline. *Principles and Practice of Structural Equation Modeling.* THE GUILFORD PRESS, 370 Seventh Avenue, Suite 1200, New York, NY 1000, 4 edition, 2016. ISBN 978-1-4625-2334-4. URL https://eli.johogo.com/Class/CCU/SEM/_Principles%20and%20Practice%20of%20Structural%20Equation%20Modeling_Kline.pdf.

[36] C. B. Kretschmer and R. Wiebe. *Liquid-Vapor Equilibrium of Ethanol–Toluene Solutions.* The Journal of Chemical Thermodynamics, 1949. doi: 10.1021/ja01173a076. URL https://webbook.nist.gov/cgi/cbook.cgi?ID=C64175&Mask=4&Type=ANTOINE&Plot=on#ANTOINE.

[37] J.-P. Lai, S. Lin, V. Lin, A. Kang, Y.-P. Wang, and P.-F. Pai. *Predicting Thermal Resistance of Packaging Design by Machine Learning Models.* Micromachines, 2025. doi: 10.3390/mi16030350. URL https://www.mdpi.com/2072-666X/16/3/350.

[38] Lehrstuhl für Wärme- und Stoffübertragung. Wärme- und stoffübertragung i/ii. 2023. URL https://www.wsa.rwth-aachen.de/go/id/hgdc.

[39] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, Cambridge, 2012. ISBN 9780511791253. doi: 10.1017/CBO9780511791253. URL https://doi.org/10.1017/CBO9780511791253.

[40] M. J. Lewis, A. Spiliopoulou, K. Goldmann, C. Pitzalis, P. McKeigue, and M. R. Barnes. *nestedcv: an R package for fast implementation of nested cross-validation with embedded feature selection designed for transcriptomics and high-dimensional data.* International Society For Computational Biology, 2023. doi: 10.1093/bioadv/vbad048. URL https://academic.oup.com/bioinformaticsadvances/article/3/1/vbad048/7117540?login=true.

[41] S. Liu, J. Li, X. Dong, and H. Chen. *Experimental study of flow patterns and improved configurations for pulsating heat pipes.* Journal of Thermal Science, 2007. doi: 10.1007/s11630-007-0056-8. URL https://link.springer.com/article/10.1007/s11630-007-0056-8.

[42] G. Łukaszewicz and P. Kalita. *Navier-Stokes Equations: An Introduction with Applications.* Advances in Mechanics and Mathematics. Springer International Publishing and Imprint: Springer, Cham, 1st ed. 2016 edition, 2016. ISBN 9783319277608. URL https://link.springer.com/book/10.1007/978-3-319-27760-8.

[43] S. M. Lundberg and S.-I. Lee. *A Unified Approach to Interpreting Model Predictions.* arXiv:1802.03888, 2017. doi: 10.48550/arXiv.1705.07874. URL https://doi.org/10.48550/arXiv.1705.07874.

[44] S. M. Lundberg, G. G. Erion, and S.-I. Lee. *Consistent Individualized Feature Attribution for Tree Ensembles.* arXiv:1802.03888, 2018. doi: 10.48550/arXiv.1802.03888. URL https://arxiv.org/abs/1802.03888.

[45] M. Mameli, M. Marengo, and S. Khandekar. *Local heat transfer measurement and thermo-fluid characterization of a pulsating heat pipe.* International Journal of Thermal Sciences, 2014. doi: 10.1016/j.ijthermalsci.2013.07.025. URL https://www.sciencedirect.com/science/article/pii/S1290072913001890.

[46] M. Mameli, M. Manzoni, L. Araneo, S. Filippeschi, and M. Marengo. *Pulsating Heat Pipe in Hypergravity Conditions.* Heat Pipe Science and Technology, 2015. doi: 10.1615/HeatPipeScieTech.2016013605. URL https://www.researchgate.net/publication/301239494_Pulsating_Heat_Pipe_in_Hyper-Gravity_Conditions.

[47] W. G. Manning and J. Mullahy. *Estimating log models: to transform or not to transform?* Journal of Health Economics, 2001. doi: 10.1016/S0167-6296(01)00086-8. URL https://www.sciencedirect.com/science/article/pii/S0167629601000868.

[48] B. Markal and K. Aksoy. *The combined effects of filling ratio and inclination angle on thermal performance of a closed loop pulsating heat pipe.* Heat and Mass Transfer, 2021. doi: 10.1007/s00231-020-02988-6. URL https://link.springer.com/article/10.1007/s00231-020-02988-6.

[49] J.-P. Melchior, G. Majer, and K.-D. Kreuer. *Why do proton conducting polybenzimidazole phosphoric acid membranes perform well in high-temperature PEM fuel cells?* Physical chemistry chemical physics, 2016. doi: 10.1039/c6cp05331a. URL https://pubs.rsc.org/en/content/articlelanding/2017/cp/c6cp05331a.

[50] J. Mukhopadhaya and D. Rutherford. *Performance analysis of evolutionary hydrogen-powered aircraft.* The International Council of Clean Transportation, 2022. URL https://theicct.org/publication/aviation-global-evo-hydrogen-aircraft-jan22/.

[51] J. Mushava and M. Murray. *Flexible loss functions for binary classification in gradient-boosted decision trees: An application to credit scoring.* Expert Systems with Applications, 2024. doi: 10.1016/j.eswa.2023.121876. URL https://www.sciencedirect.com/science/article/pii/S0957417423023783.

[52] National Institute of Standards and Technology. Thermophysical properties of fluid systems, 2026. URL https://webbook.nist.gov/chemistry/fluid/.

[53] V. S. Nikolayev. *Physical principles and state-of-the-art of modeling of the pulsating heat pipe: A review.* Applied Thermal Engineering, 2021. doi: 10.1016/j.applthermaleng.2021.117111. URL https://www.sciencedirect.com/science/article/pii/S1359431121005512.

[54] V. S. Nikolayev and M. Marengo, editors. *Pulsating heat pipes: basics of functioning and modeling. Encyclopedia of Two-Phase Heat Transfer and Flow IV.* 2018. ISBN

9789813234369. doi: 10.1142/9789813234406{\textunderscore}0002. URL https://cea.hal.science/cea-03261340v1/file/ModelingRev.vf.pdf.

[55] M. Opalski. *Numerical and Experimental Study of a Small Cryogenic Pulsating Heat Pipe.* Doctoral thesis, Wrocław University of Science and Technology and Université Paris-Saclay, 2025. URL https://wme.pwr.edu.pl/d/HGBUKOTtQKxVvBEZ0ShNuClBQSXx_XUFABWdbExYbWGFBRAg9RgNtcXccVjU3ClVBDkNGf3xdWlcU/phd-marcinopalski.pdf.

[56] M. Opalski, C. Czajkowski, P. Błasiak, A. I. Nowak, J. Ishimoto, and S. Pietrowicz. *Comprehensive numerical modeling analysis and experimental validation of a multi-turn pulsating heat pipe.* International Communications in Heat and Mass Transfer, 2024. doi: 10.1016/j.icheatmasstransfer.2024.107990. URL https://www.sciencedirect.com/science/article/pii/S0735193324007528.

[57] Optuna Contributors. *Optuna samplers.* 2018. URL https://optuna.readthedocs.io/en/stable/reference/samplers/index.html.

[58] Optuna Contributors. *sampler.py — GP Sampler Source Code (Optuna).* 2026. URL https://github.com/optuna/optuna/blob/master/optuna/samplers/_gp/sampler.py.

[59] Optuna Contributors. *sampler.py — TPE Sampler Source Code (Optuna).* 2026. URL https://github.com/optuna/optuna/blob/master/optuna/samplers/_tpe/sampler.py.

[60] K. Parmar, N. Parmar, A. K. Parwani, and S. Tripathi. *Experimental studies and machine learning approaches for thermal parameters prediction and data analysis in closed-loop pulsating heat pipes with Al2O3-DI water nanofluid.* Journal of Thermal Analysis, 2025. doi: 10.1007/s10973-024-13859-1. URL https://link.springer.com/article/10.1007/s10973-024-13859-1.

[61] S. Parthasarathi, S. Nagarajan, S. Desai, S. U. Reddy, and G. S. V. L. Narasimham. *Effect of bend radius and insulation on adiabatic section on the performance of a single closed loop pulsating heat pipe: experimental study and heat transfer correlation.* Heat and Mass Transfer, 2021. doi: 10.1007/s00231-021-03085-y. URL https://link.springer.com/article/10.1007/s00231-021-03085-y.

[62] V. M. Patel and H. B. Mehta. *Thermal performance prediction models for a pulsating heat pipe using Artificial Neural Network (ANN) and Regression/Correlation Analysis (RCA).* Sādhanā, 2018. doi: 10.1007/s12046-018-0954-3. URL https://link.springer.com/article/10.1007/s12046-018-0954-3.

[63] V. M. Patel and H. B. Mehta. *Channel wise displacement-velocity-frequency analysis in acetone charged multi-turn Closed Loop Pulsating Heat Pipe.* Energy Conversion and Management, 2019. doi: 10.1016/j.enconman.2019.05.014. URL https://www.sciencedirect.com/science/article/pii/S0196890419305618.

[64] H. Pothina and K. V. Nagaraja, editors. *Artificial Neural Network and Math Behind It.* Smart Trends in Computing and Communications, 2022.

doi: 10.1007/978-981-16-9967-2. URL https://link.springer.com/book/10.1007/978-981-16-9967-2.

[65] J. Ramousse, O. Lottin, S. Didierjean, and D. Maillet. *Heat sources in proton exchange membrane (PEM) fuel cells.* Journal of Power Sources, 2009. doi: 10.1016/j.jpowsour.2009.03.038. URL https://www.sciencedirect.com/science/article/pii/S0378775309005485.

[66] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning.* Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 2006. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://direct.mit.edu/books/oa-monograph/2320/Gaussian-Processes-for-Machine-Learning.

[67] B. Rozemberczki, L. Watson, P. Bayer, H.-T. Yang, O. Kiss, S. Nilsson, and R. Sarkar. *The Shapley Value in Machine Learning.* arXiv:2202.05594, 2022. doi: 10.48550/arXiv.2202.05594. URL http://arxiv.org/pdf/2202.05594v2.

[68] K. Saleh, M. A. El Hameedy, W. M. Mabrouk, and A. Metwally. *XGBoost regression for robust acoustic impedance prediction in the absence of density and sonic logs.* Scientific reports, 2025. doi: 10.1038/s41598-025-24727-9. URL https://www.nature.com/articles/s41598-025-24727-9.

[69] L. Scandurra, P. Alexias, and E. de Villiers. *A Framework for Initial Transient Detection and Statistical Assessment of Convergence in CFD Simulations.* arXiv:2511.22618. doi: 10.48550/arXiv.2511.22618. URL https://arxiv.org/abs/2511.22618.

[70] H. Schlichting and K. Gersten. *Boundary-Layer Theory.* Springer Berlin Heidelberg, 9 edition, 2017. ISBN 978-3-662-52917-1. doi: 10.1007/978-3-662-52919-5. URL https://link.springer.com/book/10.1007/978-3-662-52919-5.

[71] S. S. Shapiro and M. B. Wilk, editors. *An Analysis of Variance Test for Normality.* Oxford University Press, 1965. doi: 10.2307/2333709. URL https://www.jstor.org/stable/2333709.

[72] J. Snoek, H. Larochelle, and R. P. Adams. *Practical Bayesian Optimization of Machine Learning Algorithms.* University of Toronto, Université de Sherbrooke, Université de Sherbrooke, 2012. doi: 10.48550/arXiv.1206.2944. URL https://doi.org/10.48550/arXiv.1206.2944.

[73] S. Sotiriadou, E. Ntonti, D. Velliadou, K. D. Antoniadis, M. J. Assael, and M. L. Huber. *Reference Correlation for the Viscosity of Ethanol from the Triple Point to 620 K and Pressures Up to 102 MPa.* International Journal of Thermophysics, 2023. doi: 10.1007/s10765-022-03149-z.

[74] M. Stange, M. E. Dreyer, and H. J. Rath. *Capillary driven flow in circular cylindrical tubes.* Physics of Fluids, 2003. doi: 10.1063/1.1596913. URL https://pubs.aip.org/aip/pof/article/15/9/2587/450889/Capillary-driven-flow-in-circular-cylindrical.

[75] Z. Tan, Z. Cao, W. Chu, and Q. Wang. *Improvement on evaporation-condensation prediction of Lee model via a temperature deviation based dynamic correction on*

evaporation coefficient. Case Studies in Thermal Engineering, 2023. doi: 10.1016/j.csite.2023.103147. URL https://www.sciencedirect.com/science/article/pii/S2214157X23004537?via%3Dihub.

[76] The SciPy community. *scipy.stats.shapiro — SciPy v1.17.0 Manual.* 2026. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.shapiro.html.

[77] The SciPy community. *Statistical functions (scipy.stats).* 2026. URL https://docs.scipy.org/doc/scipy/reference/stats.html.

[78] O. Ubbink. *Numerical Prediction of Two Fluid Systems with Sharp Interfaces.* Doctoral thesis, Imperial College of Science, Technology & Medicine, 1997. URL https://spiral.imperial.ac.uk/server/api/core/bitstreams/ea4d6eba-d28d-4c72-b27f-7100d734b970/content.

[79] UNFCCC. *Adoption of the Paris Agreement.* 2015. URL https://unfccc.int/sites/default/files/english_paris_agreement.pdf.

[80] X. Wang, Y. Yan, X. Meng, and G. Chen. *A general method to predict the performance of closed pulsating heat pipe by artificial neural network.* Applied Thermal Engineering, 2019. doi: 10.1016/j.applthermaleng.2019.113761. URL https://www.sciencedirect.com/science/article/pii/S1359431119308610?via%3Dihub.

[81] J. Wen. *Thermal resistance modeling of oscillating heat pipes filled with acetone by using artificial neural network.* Journal of Thermal Analysis and Calorimetry, 2021. doi: 10.1007/s10973-020-10536-x. URL https://link.springer.com/article/10.1007/s10973-020-10536-x.

[82] J. M. Wooldridge. *Introductory Econometrics: A Modern Approach, Fifth Edition.* South-Western Cengage Learning, Mason, OH 45040 USA, 5 edition, 2012. ISBN ISBN-13: 978-1-111-53104-1. URL https://ia902804.us.archive.org/23/items/introductoryeconometricsamodernapproach/Introductory%20Econometrics-%20A%20modern%20Approach.pdf.

[83] XGB Contributors. Xgb parameters, 2026. URL https://github.com/dmlc/xgboost/blob/master/R-package/man/xgb.params.Rd.

[84] S. Xing, C. Zhao, J. Zou, S. Zaman, Y. Yu, H. Gong, Y. Wang, M. Chen, M. Wang, M. Lin, and H. Wang. *Recent advances in heat and water management of forced-convection open-cathode proton exchange membrane fuel cells.* Renewable and Sustainable Energy Reviews, 2022. doi: 10.1016/j.rser.2022.112558. URL https://www.sciencedirect.com/science/article/pii/S1364032122004579?via%3Dihub.

[85] L. Yang, N.-N. Nik-Ghazali, M. A. H. Ali, W. T. Chong, Z. Yang, and H. Liu. *A review on thermal management in proton exchange membrane fuel cells: Temperature distribution and control.* Renewable and Sustainable Energy Reviews, 2023. doi: 10.1016/j.rser.2023.113737. URL https://www.sciencedirect.com/science/article/pii/S1364032123005944.

[86] H. Yildirim. *The Multicollinearity Effect on the Performance of Machine Learning Algorithms: Case Examples in Healthcare Modelling.* Academic Platform Journal of

Engineering and Smart Systems, 2024. doi: 10.21541/apjess.1371070. URL https://dergipark.org.tr/en/pub/apjess/article/1371070.

[87] X. Zhang and V. S. Nikolayev. *Physics and modeling of liquid films in pulsating heat pipes*. Physical Review Fluids, 2023. doi: 10.1103/PhysRevFluids.8.084002. URL https://journals.aps.org/prfluids/abstract/10.1103/PhysRevFluids.8.084002.

A. Additional Figures

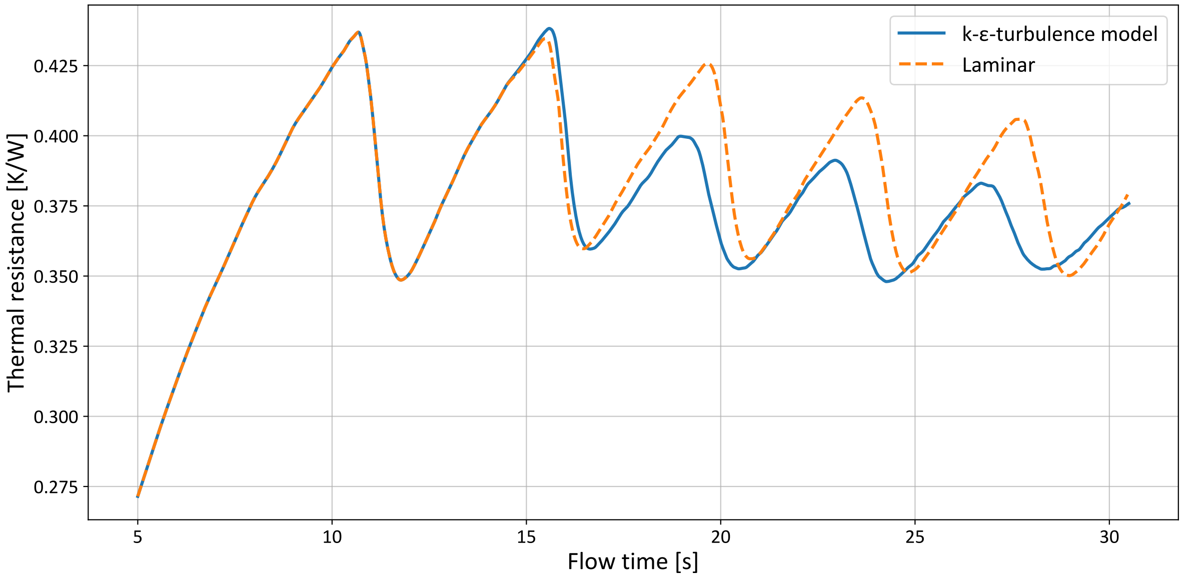

Figure A.1: Sensitivity analysis of the flow regime: comparison of predicted thermal resistance for experiment II (35W) using laminar and turbulent modeling approaches

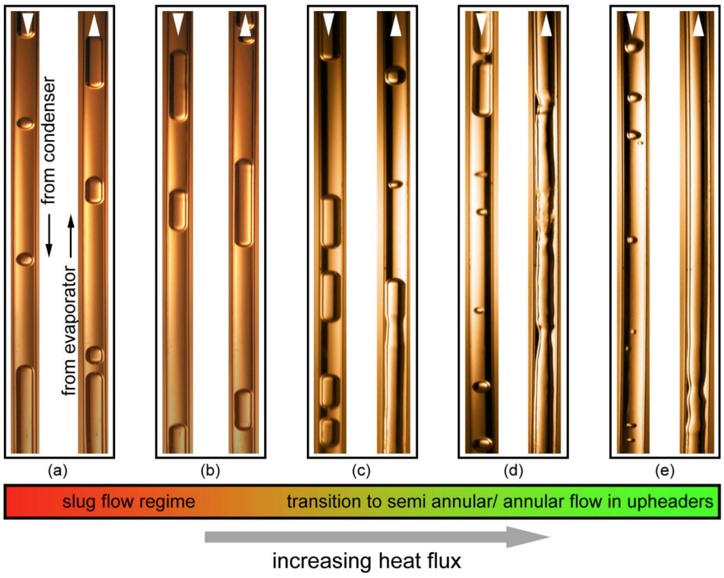

Figure A.2: Effect of heat flux on internal flow patterns inside the single loop device (experiment III) [34]

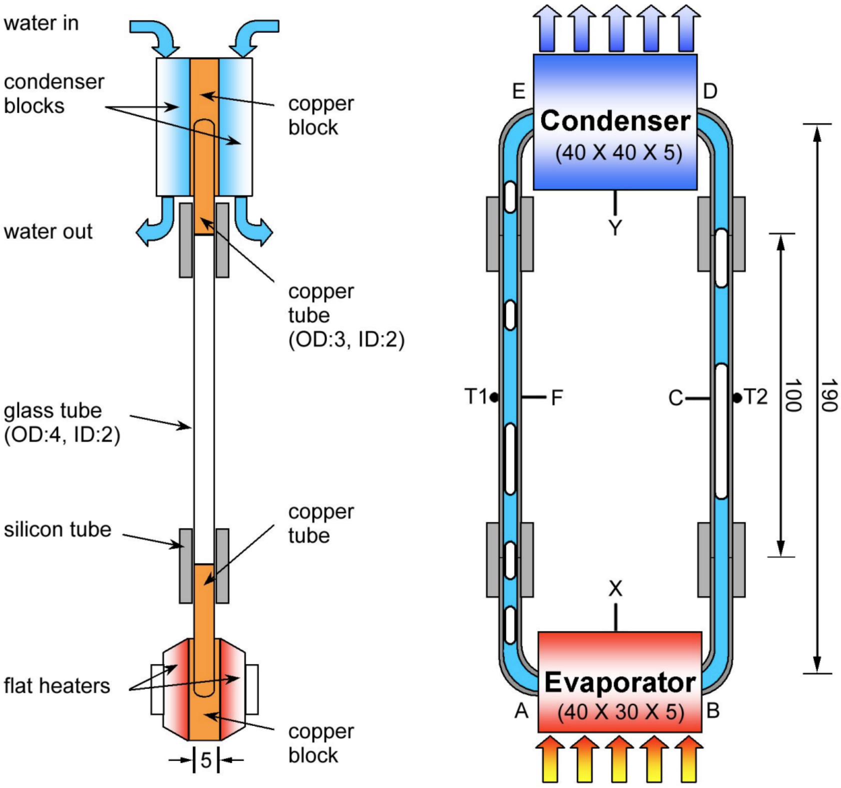

Figure A.3: Schematic details of experiment III [34]

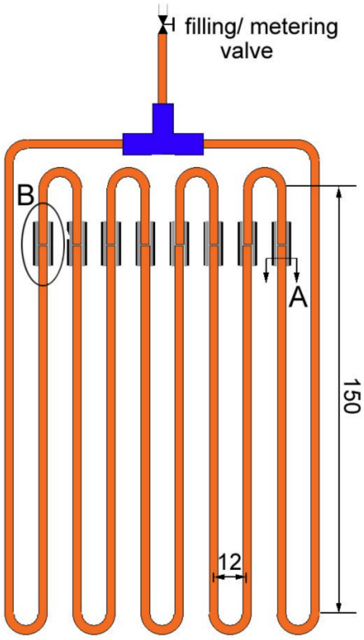

Figure A.4: Schematic details of experiment II [34]

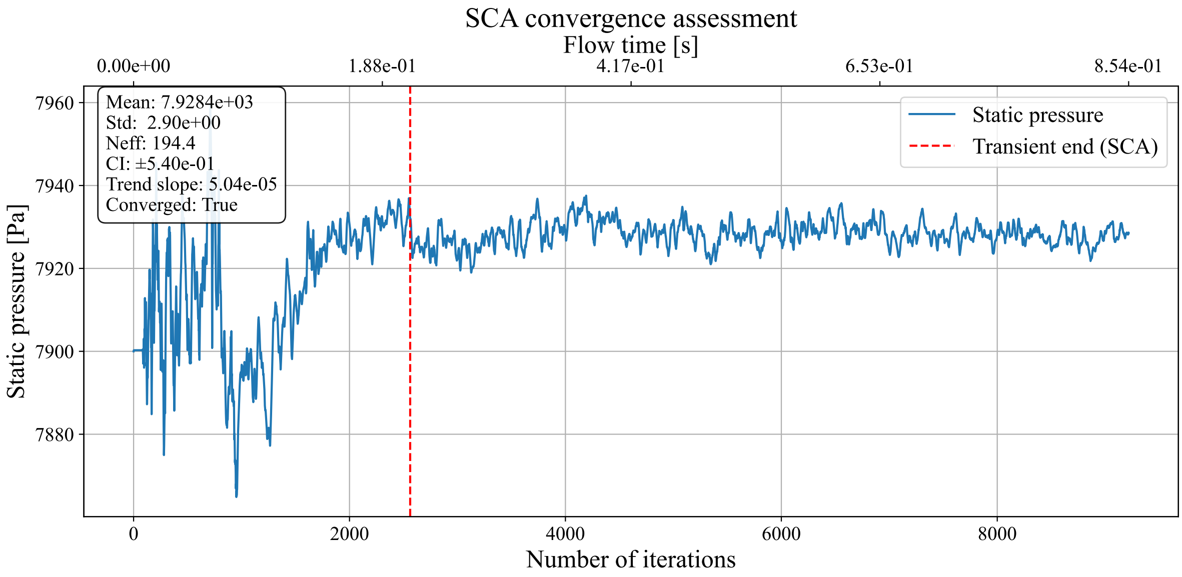

Figure A.5: Steady-state verification of the 300k-cell computational grid for experiment III

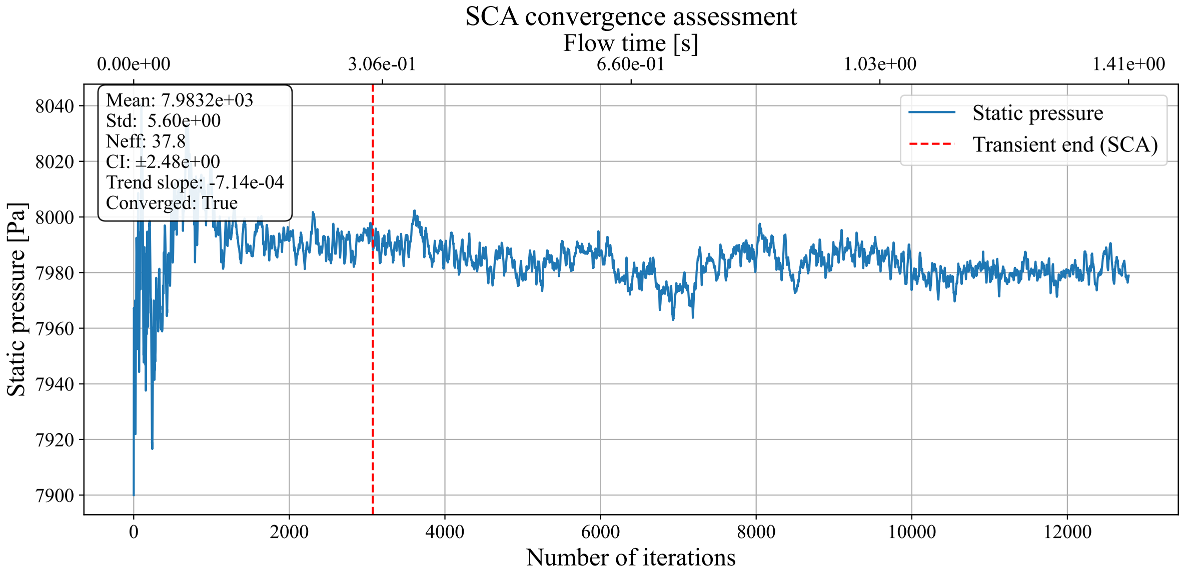

Figure A.6: Steady-state verification of the 700k-cell computational grid for experiment II

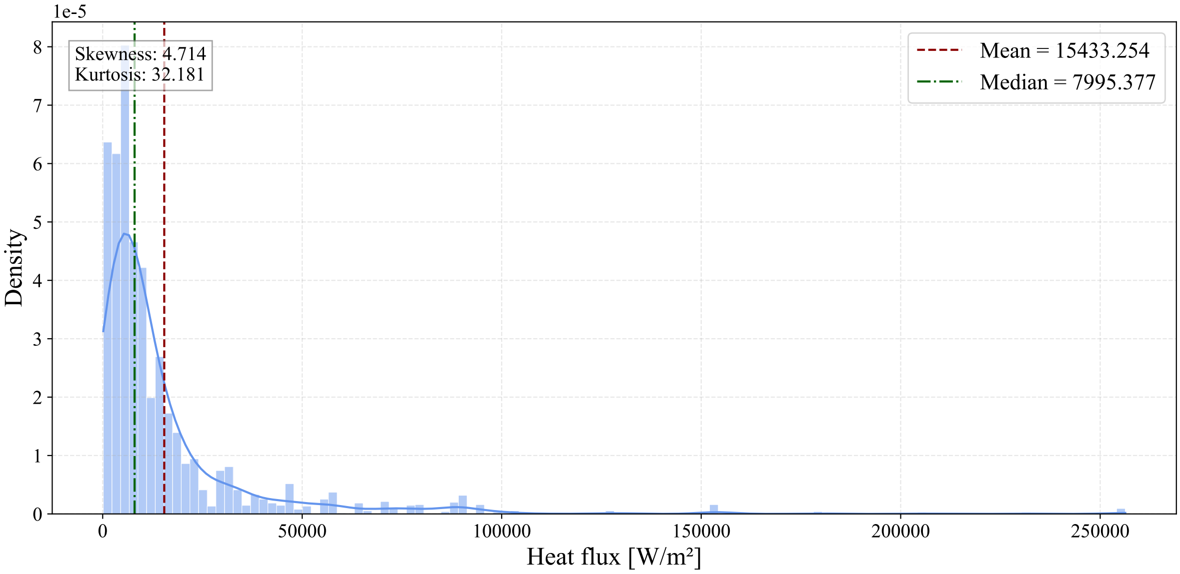

Figure A.7: Distribution of the heat flux input values

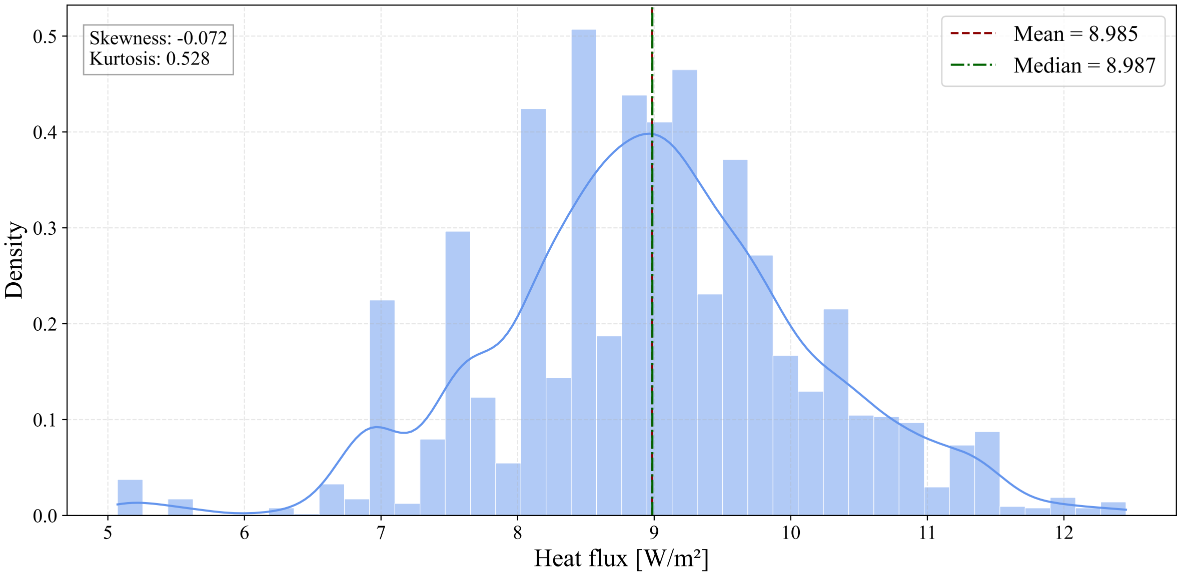

Figure A.8: Distribution of the heat flux input values on a logarithmic scale

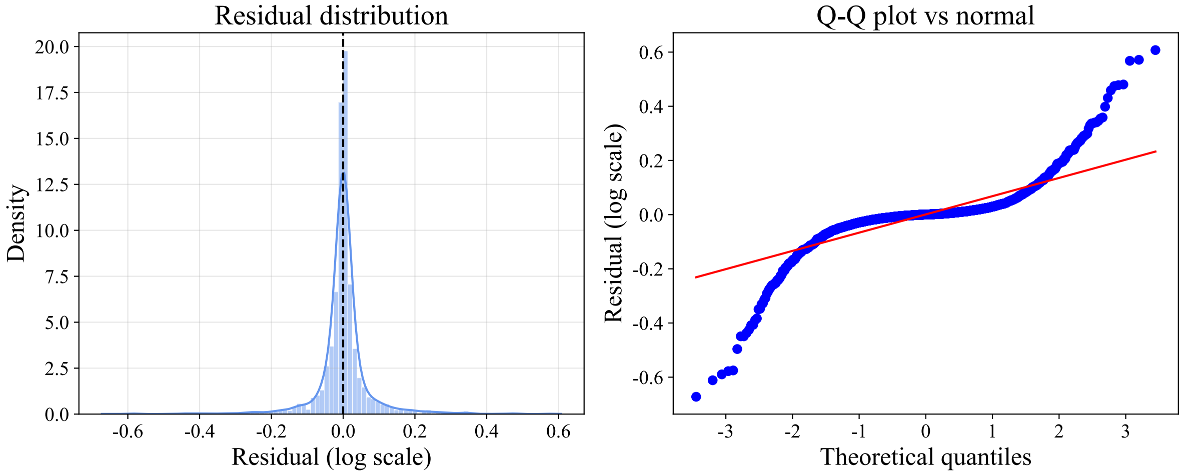

Figure A.9: Analysis of residual distribution and normality for the 70 % training dataset on a logarithmic scale

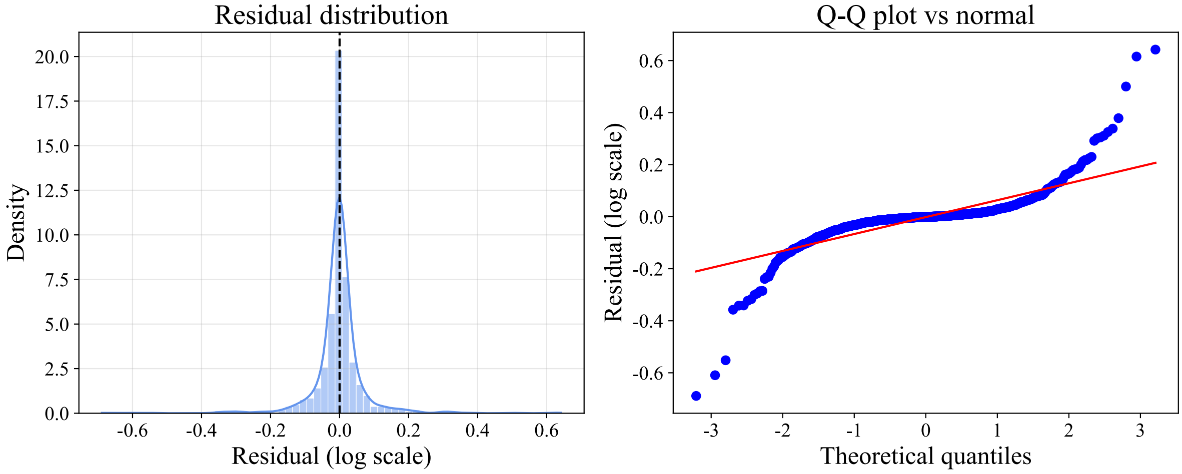

Figure A.10: Analysis of residual distribution and normality for the 30% testing dataset on a logarithmic scale

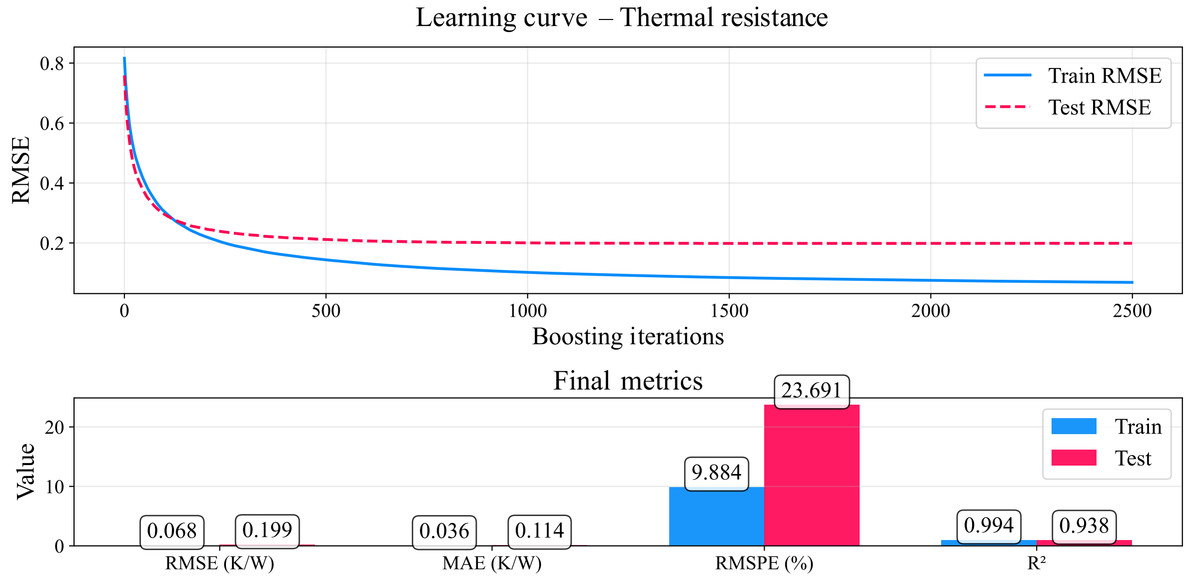

Figure A.11: XGB model performance on training and testing dataset without logarithmic transformation of thermal resistance

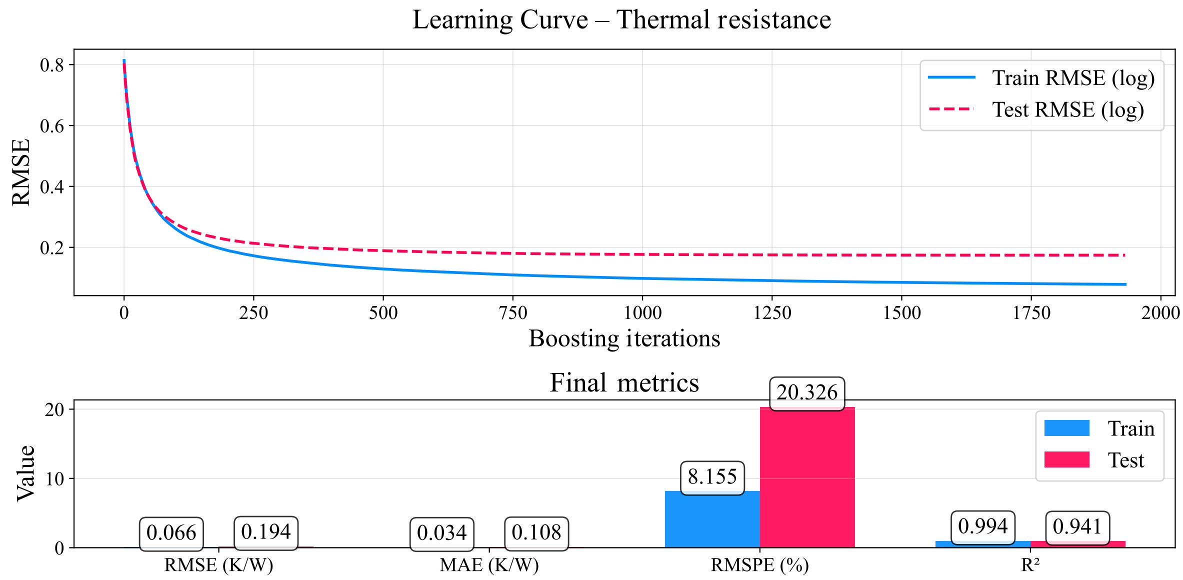

Figure A.12: XGB model performance on training and testing dataset with logarithmic transformation of thermal resistance and final metrics on original non-logarithmic scale

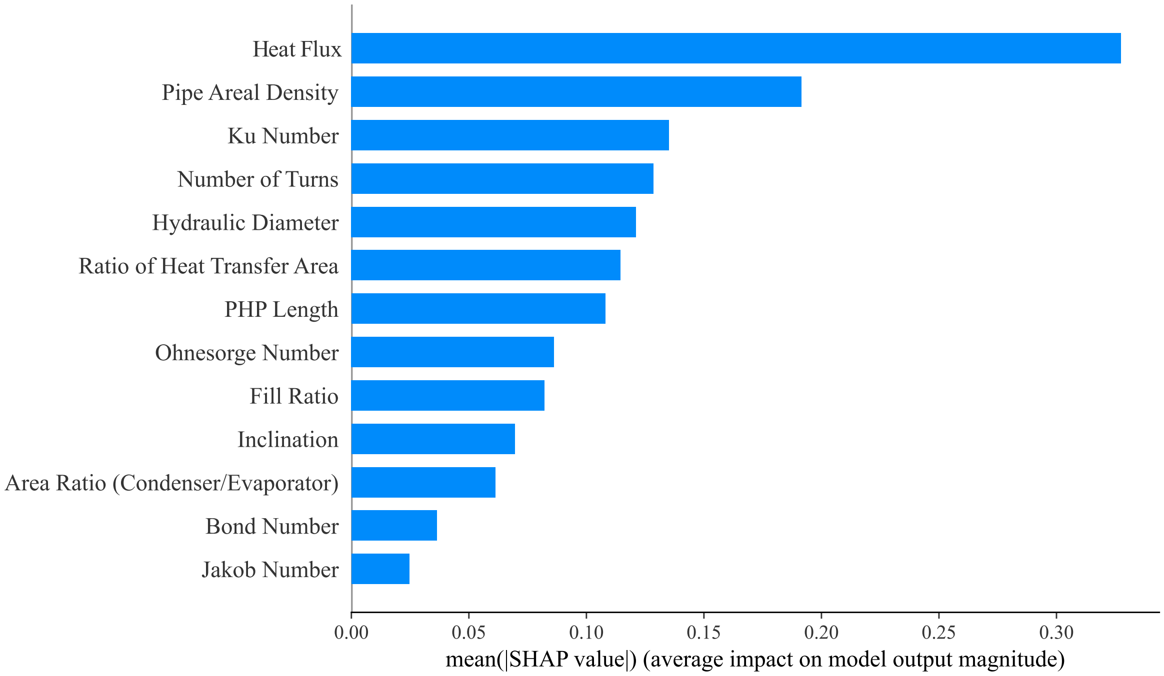

Figure A.13: Bar SHAP plot illustrating the absolute average Impact of Input Features on the predicted thermal Resistance

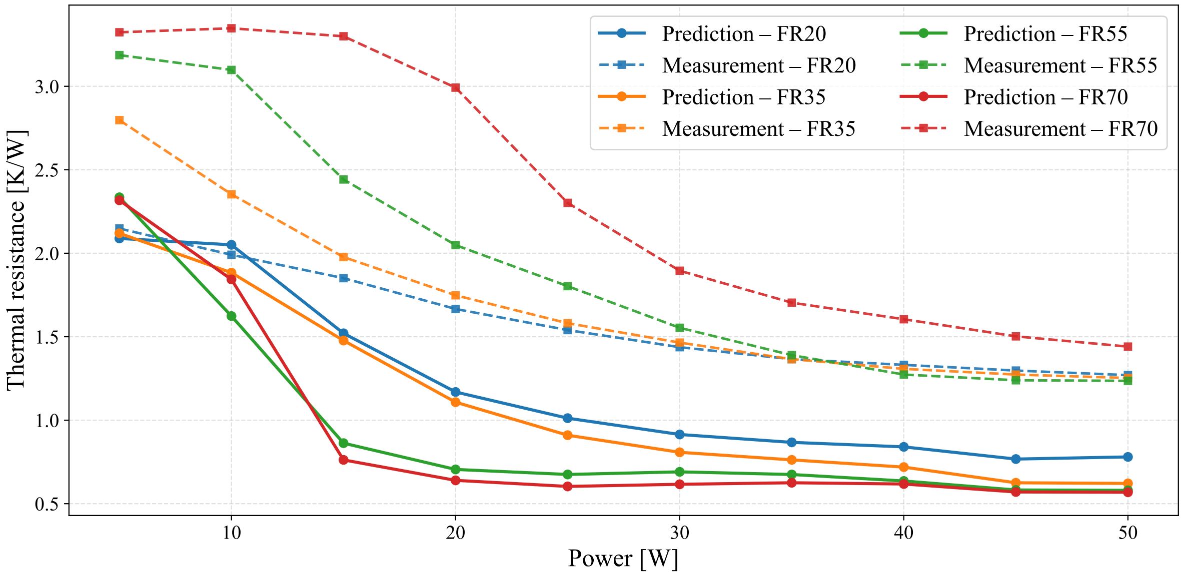

Figure A.14: Predicted vs measured thermal resistance of experiment II for different fill ratios (FR) | RMSPE = 47.1% | MAPE = 45.26%

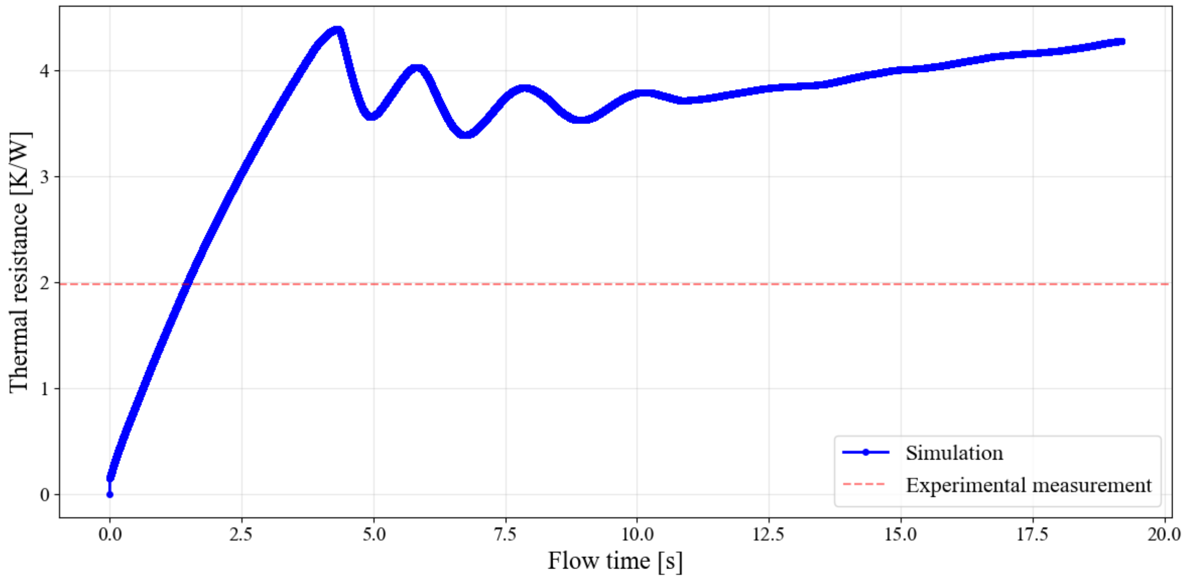

Figure A.15: Simulated thermal resistance of experiment III (32.1 W) using ethanol and a fill ratio of 60 %

B. Additional Tables

Table B.1: Publications used for building the dataset

First Author	Year	DOI
S. Khandekar	2004	10.18419/opus-1637
X.M. Zhang	2004	10.1080/08916150490246546
S. Liu	2007	10.1007/s11630-007-0056-8
Y.-H. Lin	2008	10.1016/j.applthermaleng.2007.10.019
Z. Lin	2009	10.1007/s11431-009-0194-1
M. Shafii	2010	10.1080/01457630903547636
Y. Ji	2011	10.1016/j.expthermflusci.2011.01.007
M. Mohammadi	2012	10.1115/1.4004805
J. Qu	2013	10.1016/j.apenergy.2013.02.030
B. Verma	2013	10.4236/jectc.2013.31004
P. R. Pachghare	2013	10.1016/j.proeng.2013.01.088
Y. Hu	2013	10.1016/j.ijheatmasstransfer.2013.11.031
H. Han	2014	10.1016/j.ijthermalsci.2014.04.003
M. Mameli	2014	10.1016/j.expthermflusci.2014.04.009
N. Saha	2014	10.1016/j.ijheatmasstransfer.2014.02.067
Y. Zhu	2014	10.1016/j.ijheatmasstransfer.2014.05.042
P. R. Pachghare	2014	10.1007/s12206-014-0751-9
J. Zhao	2016	10.1016/j.ijheatmasstransfer.2016.03.108
S. Shi	2016	10.1016/j.applthermaleng.2016.04.014
X. Liu	2016	10.3390/app6110321
J. Zhao	2016	10.1016/j.applthermaleng.2015.09.018
C. Y. Tseng	2016	10.1016/j.applthermaleng.2016.06.034
A. Jokar	2016	10.1007/s00231-016-1759-8
J. Wang	2016	10.1016/j.applthermaleng.2016.03.163
H. R. Goshayeshi	2016	10.1016/j.expthermflusci.2016.01.003
B. N. Mahapatra	2016	10.1016/j.applthermaleng.2016.07.137
W. Wang	2017	10.1007/s00231-016-1958-3
J. Qu	2017	10.1016/j.expthermflusci.2017.02.022

Table B.2: Publications used for building the dataset (continued)

First Author	**Year**	**DOI**
S.W. Kang	2017	10.1016/j.applthermaleng.2017.02.051
H. R. Goshayeshi	2017	10.1016/j.physe.2017.01.014
D. Bastakoti	2018	10.1016/j.ijheatmasstransfer.2017.10.075
E. Sedighi	2018	10.1016/j.ijthermalsci.2018.08.024
J. Zhao	2018	10.1016/j.expthermflusci.2018.05.026
R. Zamani	2019	10.1007/s00231-018-2418-z
A. Wei	2019	10.1016/j.ijheatmasstransfer.2019.02.021
V.M. Patel	2019	10.1016/j.enconman.2019.05.014
M. Kumar	2019	10.1080/01457632.2018.1436390
C. Czajkowski	2019	10.1016/j.applthermaleng.2019.114534
H. Ahmad	2020	10.1016/j.ijheatmasstransfer.2019.119245
Y. Xu	2020	10.1016/j.ijheatmasstransfer.2020.119727
K. Bao	2020	10.1016/j.applthermaleng.2020.115678
P. Srikrishna	2021	10.1007/s00231-021-03085-y
F. Yao	2024	10.1016/j.ijthermalsci.2024.109502

Table B.3: VOF model settings used in ANSYS Fluent 2024 R1

Type	**Value/Setting**
Phases	2 (liquid/vapor)
Surface tension force modeling	Active
Wall adhesion	Active
Evaporation-condensation model	Lee model ($\beta_v = 0.05; \beta_c = 10$)
Saturation temperature	User-defined (Antoine equation [36])
Liquid-viscosity	User-defined (temperature correlation [73])

Table B.4: CFD solver and calculation settings in ANSYS Fluent 2024 R1

Type	**Value/Setting**
Solver	Pressure-based
Velocity formulation	Absolute
Time	Transient
Pressure-velocity-coupling	PISO (correction=1)
Spatial discretizations of volume fraction	Geo-reconstruct
Adaptive timestep	yes
Global CFL-limit	1
Residual tolerance	1e-4

C. Exemplary Tree Construction

The subset of cases, seen in Tab. C.1, is hypothetical, yet it remains within the typical parameter range observed for PHPs. The algorithm is initiated with a first guess of $\hat{y}^{(0)} = 1$ for the target thermal resistance. Now taking this value as the initial guess and subtracting the actual thermal resistance of all five listed cases from this initial guess, the resulting residuals can be formulated:

Case	Heat Input (W)	Pipe Material	Working Fluid	Thermal Resistance
1	20	Copper	Water	1.00
2	40	Aluminum	Ethanol	0.85
3	60	Copper	Ethanol	0.60
4	80	Copper	Methanol	0.55
5	100	Aluminum	Water	0.20

Table C.1: Exemplary PHP dataset

$$r = \hat{y}^{(0)} - y = [0, 0.15, 0.4, 0.45, 0.8]. \tag{C.1}$$

The residuals are then clustered based on their numerical similarity in absolute terms. This process begins with the selection of a scoring metric, such as the mean squared error (MSE, Eq. 4.3), followed by the derivation of a simplified gain formulation:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{(G_L + G_R)^2}{H_L + H_R}\right] \tag{C.2}$$

where the regularization and pruning parameters λ and γ are neglected for this illustrative example. The next step is to calculate the gain for each possible split. For the first split decision, each feature can be split in several ways. For instance, the heat input has 4 potential split points, the pipe material has 1, and the working fluid has 2. In total, this would result in 10 different gain values. The split yielding the maximum gain is selected as the first split. For the sake of simplicity, only the splits of the heat input will be considered for split decisions. Splitting on heat input partitions the 5 previous residuals into two separate nodes, denoted as the left node (G_L, H_L) and the right node (G_R, H_R), respectively. For each candidate split, the gradient and Hessian sums for the left and right nodes are computed, and then the corresponding fraction of the parent node, in this case, the root node, is subtracted. When using MSE as the loss function, its gradient and Hessian are the sum of the residuals and the number of residuals in the specific node, respectively. This procedure is repeated for all 5 candidate splits. After evaluating all possible splits on the heat input, the optimal split is found at:

$$\text{Heat input} \leq 50 \text{ W} \tag{C.3}$$

with Gain ≈ 0.135. The resulting decision tree is illustrated in Fig. C.1. For the resulting

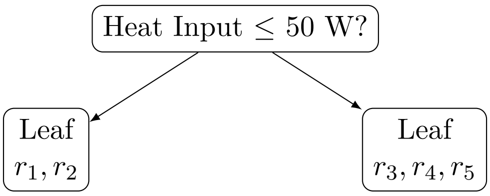

Figure C.1: First decision tree (initial guess = 1, depth = 1, $\gamma = 0$).

two nodes, each node represents a contribution to the overall prediction. This contribution is quantified by a weight, which is derived by minimizing the loss function as shown in Eq. 4.9:

$$w_j = -\frac{G_j}{H_j}. \tag{C.4}$$

Intuitively, the weight w_j determines how much the prediction should be adjusted for all samples falling into node j. By using this formulation, XGB ensures that the predicted values are updated in the direction that reduces the loss the most, effectively refining the model iteratively with each split. If the process of building the trees was terminated at this depth, these would be the corresponding weights:

- Left leaf (Heat input ≤ 50 W, cases 1 and 2):

$$G_L = 0 + 0.15 = 0.15, \quad H_L = 2, \quad w_L = -\frac{0.15}{2} = -0.075 \tag{C.5}$$

- Right leaf (Heat input > 50 W, cases 3, 4, 5):

$$G_R = 0.4 + 0.45 + 0.8 = 1.65, \quad H_R = 3, \quad w_R = -\frac{1.65}{3} = -0.55 \tag{C.6}$$

- Resulting gain:

$$G_P = 0 + 0.15 + 0.4 + 0.45 + 0.8 = 1.8, \quad \text{Gain} = \frac{1}{2}(\frac{0.15^2}{2} + \frac{1.65^2}{3} - \frac{1.8^2}{5}) \approx 0.135 \tag{C.7}$$

Repeating the same split decision logic for the resulting two nodes, the tree results in Fig. C.2 As a result, case 1 falls into the leftmost leaf of the tree, while case 2 occupies the leaf immediately adjacent to it. These leaves contain the exact thermal resistance values for the respective cases. On the right side of the first split, the leaf corresponding

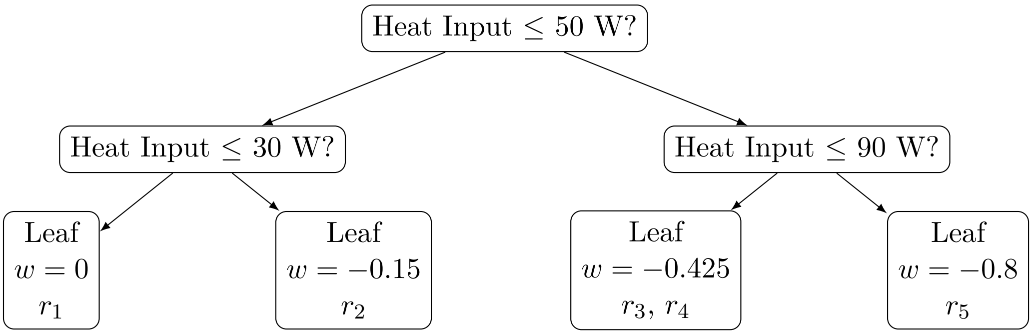

Figure C.2: First decision tree (initial guess = 1, depth = 2, $\gamma = 0$).

to values between 51 and 90 W contains case 3 and case 4, whereas the leaf for Heat input greater than 90 W contains case 5, with a thermal resistance of 0.20, appearing at the far right of the sketch.

With a learning rate $\eta = 1$, the resulting predictions of the XGB model for the thermal resistances after the first tree iteration are:

$$\hat{y}^{(1)} = \hat{y}^{(0)} + \eta\, w_j = \begin{cases} 1.00 & \text{if Heat Input } \leq 30 \text{ W}, \\ 0.85 & \text{if } 30 < \text{Heat Input } \leq 50 \text{ W}, \\ 0.575 & \text{if } 50 < \text{Heat Input } \leq 90 \text{ W}, \\ 0.20 & \text{if Heat Input } > 90 \text{ W}. \end{cases}$$

The second tree is subsequently trained on the residuals remaining from the first tree. Based on the updated predictions, the residuals for iteration $t = 2$ are calculated as:

$$g_i = \hat{y}^{(1)} - y_i = [0, 0, -0.025, 0.025, 0]$$

Since the first tree has already made substantial corrections, the residuals are small, and the second tree will make only minor adjustments. The optimal split for the second tree iteration is found at

$$\text{Heat Input} \leq 70 \text{ W}, \tag{C.8}$$

with a very small gain of approximately 0.000521, indicating that only minor residual structure remains in the data. The leaf weights are again computed according to:

- Left leaf (Heat input $\leq$ 70 W, Cases 1,2,3):

$$G_L = 0 + 0 - 0.025 = -0.025, \quad H_L = 3, \quad w_L = -\frac{-0.025}{3} \approx 0.0083 \tag{C.9}$$

- Right leaf (Heat input $>$ 70 W, Cases 4,5):

$$G_R = 0.025 + 0 = 0.025, \quad H_R = 2, \quad w_R = -\frac{0.025}{2} = -0.0125 \tag{C.10}$$

Thus, the corresponding predictions after the second tree are: The very small magnitude of the leaf weights confirms that most of the predictive structure has already been captured by the first tree, and the second tree mainly serves to perform minor corrective adjustments to the model predictions. This illustrates the incremental boosting process of XGB, where each subsequent tree corrects the residuals left by the previous trees, improving prediction accuracy step by step. The final predictions of the XGB algorithm after two tree iterations are shown in Table C.2.

Case	Heat Input (W)	w	$\hat{y}^{(2)}$
1	20	0.0083	1.0083
2	40	0.0083	0.8583
3	60	0.0083	0.5833
4	80	-0.0125	0.5625
5	100	-0.0125	0.1875

Table C.2: Updated predictions after a second tree iteration.

D. Shapley Values

Shapley values have been introduced by L. S. Shapley in the context of cooperative game theory. They quantify how much a specific player contributes to a total outcome relative to other players [67]. For the ML context, the players represent the input features, and the outcome is the predicted thermal resistance in this case. Consider a prediction with the input features heat flux, Ohnesorge number, and the working fluid (WF), resulting in a prediction of $R_{th} = 1$. Each coalition of the input features has a specific outcome as shown in Tab. D.1. The Shapley value is the weighted average of these marginal contributions.

Table D.1: Coalitions of input features and marginal contributions of the heat flux to the predicted thermal resistance

Coalition	Predicted R_{th}	Marginal contribution of heat flux
$\emptyset$	0.40	0.30
Oh	0.50	0.20
WF	0.30	0.50
Oh, WF	0.70	0.30
q'', Oh	0.70	–
q'', WF	0.80	–
q'', Oh, WF	1.00	–

The weight is calculated as the probability of a feature joining a coalition in a particular order:

$$\text{Weight} = \frac{|S|!(n - |S| - 1)!}{n!}$$

where S is the set of features already in the coalition, and n is the total number of features. The marginal contribution can also be written as:

$$val(S \cup \{i\}) - val(S). \tag{D.1}$$

For the heat flux in a 3-feature prediction ($n = 3$):

$$\phi_1 = \frac{1}{3} \cdot 0.3 + \frac{1}{6} \cdot 0.2 + \frac{1}{6} \cdot 0.5 + \frac{1}{3} \cdot 0.3 \approx 0.32$$

$\phi_1 = 0.32$ is the contribution of heat flux to the model's prediction of thermal resistance. In the context of machine learning models, Shapley values can be used as a measure of each feature's contribution to the model's output. This is important for identifying which parameters have a significant impact on the predictions.

In tree-based models such as XGB, there are no separate, independently evaluable contributions per feature in the classical sense. Instead, Tree SHapley Additive exPlanations (SHAP), an algorithm developed by S. Lundberg et al. [44], computes the marginal contribution of a feature by comparing two quantities for each relevant coalition. Firstly, the model's prediction when the feature value is known, and secondly, the expected prediction when the feature is masked, which is obtained by averaging over the possible values the feature could take according to a background distribution, which is the fraction of the other training cases that go left/right at that split. This is achieved by pushing the current observation down the trees while maintaining two parallel execution paths at splits where a feature is considered for inclusion in the coalition. When the feature at a decision split is considered known in the current coalition, the path follows the deterministic branch determined by the actual feature value of the observation. When the feature is considered masked so that it is not part of the current coalition, the probability mass is split at that node according to the fraction of background samples that would go left or right. At the leaf node, the algorithm rewinds upwards until a new feature is seen in the current. Now, the value that the tree would predict with this feature known and masked is evaluated, so the contribution of the feature to the current coalition can be calculated. This is done without explicitly enumerating the 2^{n-1} coalitions by making use of the tree structure. This recursive traversal ensures that the Shapley weights are implicitly respected through the depth and branching structure of the trees. The mathematical definition is given by:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[f_x(S \cup \{i\}) - f_x(S) \right]. \tag{D.2}$$

In the case of TreeSHAP, at each masked split of a new feature i, the coalition S corresponds to the set of features that were already revealed by the decision splits encountered from the root down until this split. Here, M denotes the total number of unique features encountered on the path evaluated for this observation. The quantity $f_x(S \cup \{i\})$ represents the model's prediction obtained by continuing deterministically from this node onward using the actual observed value of feature i, whereas $f_x(S)$ is the expected prediction when i remains masked at this node, computed by averaging over the background distribution. The difference $f_x(S \cup \{i\}) - f_x(S)$ is therefore the local marginal contribution of revealing feature i exactly at this split, weighted by the probability mass that reached this node. For a detailed description of the algorithm, the reader is referred to [44, 43].