# Journal Pre-proof

## Hybrid modelling of heat transfer systems: Combining physics-based and data-driven approaches for improved prediction and extrapolation

N. Nazemzadeh, J. Loyola-Fuentes, G. Righetti, E. Diaz-Bejarano,
S. Mancin, F. Coletti

PII: S3050-5852(26)00004-2
DOI: https://doi.org/10.1016/j.aitf.2026.100030
Reference: AITF 100030

To appear in: *AI Thermal Fluids*

Received date : 9 October 2025
Revised date : 16 December 2025
Accepted date : 28 January 2026

Please cite this article as: N. Nazemzadeh, J. Loyola-Fuentes, G. Righetti et al., Hybrid modelling of heat transfer systems: Combining physics-based and data-driven approaches for improved prediction and extrapolation, *AI Thermal Fluids* (2026), doi: https://doi.org/10.1016/j.aitf.2026.100030.

# Hybrid Modelling of Heat Transfer Systems: Combining Physics-Based and Data-Driven Approaches for Improved Prediction and Extrapolation

N. Nazemzadeh[1], J. Loyola-Fuentes[1], G. Righetti[2], E. Diaz-Bejarano[1], S. Mancin[2,3] and F. Coletti[1,3*]

[1]Hexxcell Ltd., 1 Lyric Square, W6 0NB, London, United Kingdom

[2]University of Padova, Department of Management and Engineering, S.lla San Nicola, 3, Vicenza, 36100, Italy

[3]Department of Chemical Engineering, Brunel University London, Kingston Lane, Uxbridge, UB8 3PH, United Kingdom

[*]Corresponding author: f.coletti@hexxcell.com

## Abstract

Hybrid machine learning-assisted modelling techniques have gained increasing attention recently in many engineering fields. This is due to the challenges associated with pure first-principles and data-driven models, as the former requires deep phenomenological understanding and might become infeasible to describe a complex system with, and the latter needs extensive high-quality data and, more importantly, extrapolates poorly compared to its first principles counterparts. The integration of the two techniques in a framework will result in an integrated approach that benefits from the two realms by strengthening extrapolation capabilities, higher prediction accuracy, and less data demanding and more data-efficient. In this study, a systematic hybrid modelling framework is developed, allowing for the integration of mechanistic models and machine learning algorithms in parallel and series for modelling heat transfer systems to predict a desired target variable, as long as the system is not of a dynamic nature. The framework is developed according to a previous study that enabled the use of machine learning models for such systems. The application of the hybrid modelling framework in this study is demonstrated on the prediction of the condensation heat transfer coefficient in a microfin tube. A laboratory-scale dataset of 5,708 datapoints is used for the validation of the developed framework. The validation of the model has been carried out in two different scenarios, both assessing the general prediction and extrapolation capabilities of the developed models in comparison with pure mechanistic and pure machine learning models. The hybrid models, series and parallel, outperform the mechanistic model by approximately 60% more accurate predictions and the machine learning model by 25%, while interpolating. More importantly, while extrapolating, the hybrid models showed approximately 50% more accurate predictions compared to pure machine learning and 27% more accurate compared to the mechanistic model.

## 1 Introduction

Traditionally, first-principles and mechanistic models have been broadly developed to describe complex systems, processes, and phenomena mathematically. These models are most often developed by a combinatorial formal analysis, including experimentation, data observation, and theoretical analysis. First-principles and mechanistic models are data-efficient tools that allow for reliable predictions within the search space of the systems, as long as they are based on accurate databases. However, developing such models requires a deep fundamental understanding of the system and, in most cases, becomes a time-consuming task. More importantly, in more complex systems, it is rather impossible to develop such models due to their complex nature. Accordingly, in such situations, data-driven approaches have been explored as a counterpart to provide more effective solutions for a variety of problems.

Data-driven models have recently received significant attention considering the challenges associated with first-principles and mechanistic models, and more importantly, due to rapid technological advancements in computer science, data acquisition, and data availability. These models heavily rely on data to reveal underlying correlations among different components in a system, necessitating large, high-quality datasets that in most engineering applications are challenging to collect. For instance, the process industry typically operates with low variation in operating conditions around specific setpoints due to production targets, safety constraints, and regulatory compliance. This poses a significant challenge when employing data-driven approaches for such conditions, as the variations in the data are minimal. Hence, data-driven models will face a challenge in exploring correlations within a proper search space. Consequently, the reliability of these models beyond the operational envelope can be questionable. On the other hand, laboratory-scale experiments offer more flexibility, allowing controlled variations of process variables. However, generating sufficiently large datasets under such conditions remains time-consuming and resource-intensive. Additionally, most data-driven models do not allow for physical interpretation of the predictions, and they are typically associated with poorer extrapolation capabilities compared to their first-principles counterparts. Nonetheless, their requirement for less domain knowledge and fewer assumptions, and high prediction accuracy make these models an attractive candidate for complex systems. One of the most popular classes of data-driven models is machine learning, which allows for the construction of algorithms capable of learning underlying features of a dataset. Another class of models has been introduced more recently by integrating data-driven and mechanistic approaches, called hybrid models, which inherit the advantages of both approaches and are increasingly gaining attention due to their applicability in various subjects.

In the following, an introduction to hybrid modelling and mechanistic modelling challenges in heat transfer systems is provided, justifying the application of such a modelling approach in the field of heat transfer. This study focuses on developing a hybrid machine learning-assisted modelling framework to assist the mechanistic models in heat transfer systems for a more accurate prediction of the desired target variable. The application of such a framework is agnostic to the mechanistic model combined as long as the system of study does not have dynamic characteristics. The application of such modelling frameworks is validated against laboratory-scale experimental data of condensation in microfin tubes for the estimation of the heat transfer coefficient. The paper demonstrates that the hybrid model configurations show more accurate interpolation and extrapolation compared to pure mechanistic and machine learning models.

### 1.1 Hybrid Modelling in Process Systems Engineering

Challenges associated with first-principles and data-driven models led to the development of a relatively recent modelling technique called hybrid modelling. This technique integrates data-driven approaches with mechanistic models for problems for which neither of the techniques is sufficient to model a complex system [1]. The integration of the two methods allows one to benefit from the features of both modelling realms. Therefore, in principle, a hybrid model offers accurate predictions both inside and outside of the search space, while it does not require as extensive data as data-driven models, and most certainly does not need a deep understanding of the system and its underlying phenomena compared to a fully first-principles/mechanistic model. In other words, these models provide a trade-off between *a priori* knowledge and data requirement, prediction accuracy, interpretability, and model scalability.

With the recent advancements in machine learning techniques and their computational efficiency, hybrid machine learning-assisted modelling techniques are becoming an attractive choice over others [2]. However, one must note that selecting the appropriate modelling approach for a complex system depends on several factors, such as 1) availability of first-principles models, 2) reliability of those models, 3) choice of the machine learning model, and 4) complexity level of the integration for the machine learning and mechanistic components. Figure 1 represents various modelling techniques that can be employed from a systems engineering perspective. Since machine learning (Model A) and first-principles models (Model E), the two ends of the spectrum, have already been discussed in detail, in the following, we focus on summarising the features of each hybrid model configuration and their current application examples in the literature, with a focus on Process Systems Engineering (PSE), as summarised in Table 1.

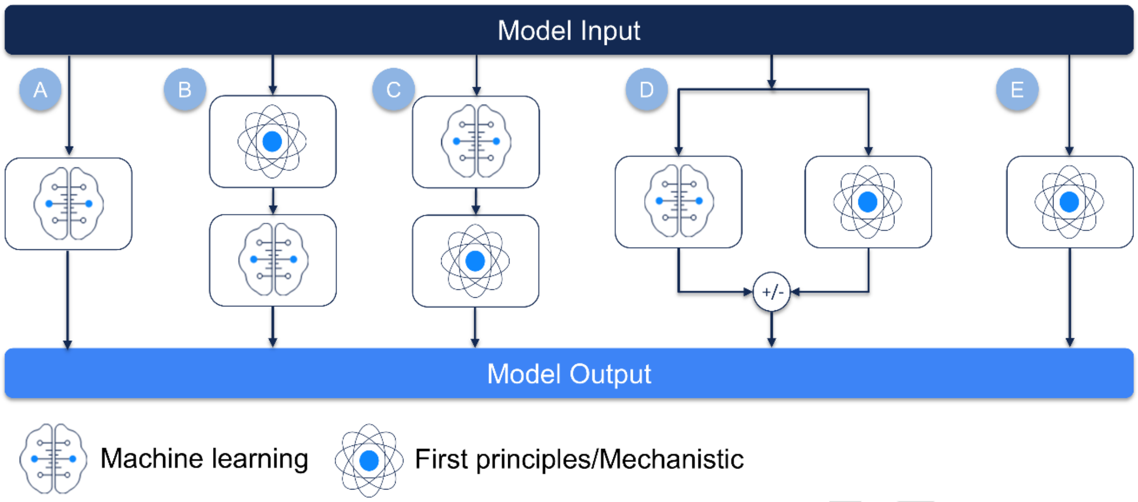


**Figure 1:** Schematic representation of modelling approaches: (A) Machine learning, (B) Hybrid series with machine learning as the main predictor, (C) Hybrid series with first principles as the main predictor, (D) Hybrid parallel, and (E) first-principles model.

Compared to a pure machine learning model, Model B offers to incorporate *a priori* knowledge into the machine learning component in the form of soft sensors, while the main predictor remains the machine learning model. Soft sensors are model-based information that is not measurable by instrumentation. This configuration has been used in several applications in the field of Chemical and Process Engineering. A hybrid model using artificial neural networks was employed in 1999 for the industrial Nylon-6,6 polymerisation process in a twin-screw extruder reactor, predicting the relative viscosity of the product as a key physical property. The prediction accuracy improved by approximately 50% using the hybrid configuration compared to the empirical-based mechanistic model, resulting in optimising the process condition to increase production by 20% [3]. In 2001, this model configuration was applied to predict pulp delignification of an industrial pulp mill, resulting in a more accurate model compared to a pure neural network model [4]. A few years later, Model B was applied to an industrial reactive distillation column for Epichlorohydrin and resulted in lowering the dissolved organic carbon in the bottom product of the column, which was achieved by reducing the alkalinity of the bottom product by almost 33% [5].

Another hybrid model configuration that has attracted great attention in several applications is Model C. This configuration allows the machine learning model to be trained for predicting certain inputs of the mechanistic model that are not directly measurable (e.g., parameters) or difficult to determine with domain knowledge. Thereafter, the first-principles/mechanistic model performs the main prediction of the target variable by using the measured input variables and the output of the machine learning model. The training can be carried out with two different approaches: 1) A pre-trained machine learning model and 2) end-to-end training of the ML model, directly predicting the outputs of the combined model [6]. One limitation of this hybrid model is that the end-to-end training usually requires the mechanistic model to be differentiable. Only under such conditions can both models be optimised together; otherwise, for non-differentiable mechanistic models, other techniques must be employed for optimisation [6].

In 2006, the application of such a configuration was successfully demonstrated by integrating an artificial neural network or ANN (estimating kinetic parameters) with mass balances of the Acetobacter xylinum cultivation in a stirred tank reactor, resulting in an excellent agreement with cultivation experimental data [7]. This configuration was also applied for on-line monitoring of Lipase production by Candida rugosa, which outperformed a fully mechanistic model [8]. More recently, the application of Model C has found its way into particulate processes with a focus on crystallisation and flocculation [2,9]. In these studies, an end-to-end training of the model was developed to predict the dynamics of the particle size distribution (PSD). The developed models accurately predicted the particle size distributions using an ANN combined with population balance equations. Moreover, in a further study, computational chemistry calculations were integrated with the developed hybrid model for flocculation, which allowed for simpler ANN model components (less complex architecture) compared to the previously developed hybrid model [10].

Model D (parallel configuration) is employed when first principles are capable of describing the system's behaviour; however, they are prone to errors. Hence, the machine learning component can learn to correct the predictions by an additive/corrective error term. In 1994, the parallel configuration was applied to model the fermentation of Penicillin in a fed-batch bioreactor [11] by integrating a Radial Basis Function Network (RBFN) with the fermentation dynamics, which resulted in accurate dynamic prediction of the process states compared to a pure data-driven and pure mechanistic approach. Another study in 2012 applied a parallel configuration for controlling the Cobalt Oxalate synthesis process, combining mass balances (i.e. population balance equations) with partial least squares (PLS) models and illustrated a successful control loop for regulating end-point particle size distribution. The study also compared the prediction of the PSD with the first principles and PLS only, where the hybrid model predictions showed more stability and higher accuracy by harnessing the extrapolating capabilities of the population balance equations [12].

**Table 1:** Summary of the studies on hybrid modelling approaches in the field of Process Systems Engineering

| Study | Application | Data-driven model | No. layers | Mechanistic model | Model |
|---|---|---|---|---|---|
| **Nascimento, Guidici, Scherbakoff, 1999** [3] | Nylon-6,6 extruder process | ANN | 1 | Mass balances | B |
| **Aguiar and Filho, 2001** [4] | Pulp delignification | ANN | 1 | Mass balances | B |
| **Chen et al. 2003** [5] | Reactive distillation for Epichlorohydrin | ANN | 1 | Mass balances | B |
| **Zuo et al., 2006** [7] | Airlift reactor of Acetobacter xylinum | ANN | 1 | Mass balances | C |
| **Boareto et al., 2007** [8] | Lipase production by Candida rugosa | ANN | 1 | Mass balances | C |
| **Nielsen et al. 2021** [2] | Lactose and pharmaceutical crystallisation, Silica particles flocculation | ANN | 4 | PBE | C |
| **Nazemzadeh et al., 2021** [9] | Silica particle flocculation | ANN | 4 | PBE | C |

| | | | | | |
|---|---|---|---|---|---|
| **Nazemzadeh, 2022** [10] | Silica particle flocculation | ANN | (2, 3, 4) | PBE and nano-scale interactions between particles | C |
| **Thompson and Kramer, 1994** [11] | Fed-batch bioreactor for Penicillin | RBFN | 1 | Mass balances | D |
| **Zhang et al., 2012** [12] | Cobalt oxalate synthesis process | PLS | N/A | PBE | D |

## 1.2 Mechanistic Modelling Challenges in Heat Transfer Systems

Designing efficient, compact, and cost-effective heat exchangers is a must in process engineering, and up to today, it has always relied on the accuracy and predictive capabilities of traditional mechanistic models developed since the 1960s. In fact, after World War II, a huge effort was put into developing design methods for simple (single-phase) and complex (two-phase and multi-phase) heat transfer systems. Kern [13] was one of the pioneers in the field, and his handbook is still considered one of the bibles for heat exchanger design in the process industry. More recently, the work done by Prof. G. Hewitt culminated in one of the most comprehensive handbooks on heat exchanger design [14] which is still integrating and updating novel designs and methods.

These traditional methods are based on the estimation of heat transfer coefficients through mechanistic models, which commonly use specific dimensionless numbers and are regressed on experimental databases. A classic example of these models is the famous and still widely used Dittus-Boelter [15] correlation. This correlation is based on a large database comprising hundreds of experiments collected during single-phase turbulent heat transfer inside several tubes. The model takes full advantage of the heat and mass analogy, which allows theoretically correlating the Nusselt number to be a power function of the Reynolds number and Prandtl number. Then, a simple regression permitted the development of one of the most accurate, widely used, and long-lived heat transfer correlations. The typical accuracy in the prediction of the heat transfer coefficient is in the range of 10-30% [16,17].

The predictive capabilities of such traditional mechanistic methods depend upon the accuracy of the experimental database, but also rely on the understanding of the heat and mass transfer mechanisms underpinning the phenomenon. Thus, the success of the Dittus-Boelter correlation can be attributed to two main factors: the large and accurate database of experimental data points and the deep knowledge of the theoretical mechanisms governing heat transfer.

However, in two-phase systems, the complexity of the phenomena increases exponentially, and even after decades of excellent research activities, the theoretical knowledge on, for example, condensation and boiling, is still incomplete. For this reason, semi-empirical approaches most of the time fail to accurately predict the heat transfer performance of such systems.

In any case, several successful attempts to develop accurate and robust semi-empirical correlations have been proposed in the open literature. Considering condensation heat transfer inside tubes, Righetti et al. (2016) [18] reviewed several models proposed either for boiling or condensation heat transfer inside smooth and microfin tubes. The ones proposed by Kim and Mudawar, (2014) [19] exhibited the best performance in both condensation and boiling inside smooth pipes. Regarding microfin tubes, the model proposed by Cavallini et al. (2009) [20] for condensation showed the best agreement with the experimental database. This model is based on more than 40 years of research done by the Cavallini group and took advantage of the experimental activities carried out and of the previous models developed and validated on thousands of datapoints. In any case, the absolute average deviation of these models always ranges between 15%-30%.

As already described by Loyola-Fuentes et al. (2024) [21], the complexity of these phenomena and the plethora of parameters that play a role in determining the final two-phase behaviour imply an incredible difficulty in identifying the correlations among those variables. This paper aims to provide a novel framework for heat transfer modelling, trying to combine mechanistic and machine learning methodologies to maximise the accuracy given by machine learning tools while maintaining the predictability guaranteed by mechanistic methods.

### 1.3 Machine Learning and Hybrid Approaches in Heat Transfer Systems

The challenges in mechanistic modelling of heat transfer systems described in the previous section have led several researchers to rely on data-driven and, more precisely, machine learning-assisted modelling of complex heat transfer systems. One must note that the application of hybrid modelling approaches illustrated in Figure 1 in the field of heat transfer has not been as widely recognised as generally seen in the PSE community. Most of the efforts have focused thus far on developing purely machine learning models or Physics-Informed Machine Learning (PIML). Although PIML models can be categorised as a form of hybrid model, but the ML component is dominating the overall model performance, as the mechanistic model only imposes some constraints on the ML model in the loss function. The table below summarises some of the ML-assisted modelling applications in the field of heat transfer.

**Table 2:** Summary of the studies on ML-assisted modelling approaches in the field of Heat Transfer

| Study | Application | Data-driven model | Mechanistic model | Model Configuration |
|---|---|---|---|---|
| **Khosravi et al. 2018 [22]** | Modelling pressure drop during evaporation of R407C | ANN and SVR | N/A | A |
| **Hobold and da Silva 2019 [23]** | Visualisation-based quantification of nucleate boiling heat flux | PCA, CNN, and MLP | N/A | A |
| **Kwon et al. 2020 [24]** | Modelling local convective HTC of a cooling channel | RF regressor | N/A | A |
| **Peng et al. 2020 [25]** | Modelling heat conduction in complex geometries | CNN | N/A | A |
| **Souayeh et al. 2021 [26]** | Modelling friction factor and Nusselt number of flow in circular tube | ANN | N/A | A |
| **Loyoal-Fuentes et al. 2022 [27]** | Flow pattern classification in heat pipes | KNN, RF, and MLP | N/A | A |
| **Jayaweera et al. 2022 [28]** | Thermal performance of direct contact cooling towers | ANN | Restrep Reyes model | C |
| **Loyoal-Fuentes et al. 2024 [21]** | Modelling condensation heat transfer coefficient in microfin tubes | ANN and RF | N/A / Dimensionless numbers | A / B |
| **Lee et al. 2025 [29]** | Prediction of condensation heat transfer coefficient | XGBoost | Degradation Nusselt model | Physics-Informed |

| | | | | |
|---|---|---|---|---|
| **Furlong et al. 2025 [30]** | Modelling of critical heat flux for boiling systems | ANN | Biais and Bowring correlations | D* |

An example of ML modelling in heat transfer is the study by Khosravi et al. (2018) [22], in which they developed models using ANN and SVM to predict the pressure drop of R407C. Their analysis showed that ANN outperforms the SVM by having a $R^2 = 0.998$. Hobold and da Silva (2019) [23] applied various models, including PCA, CNN, and MLP, to infer the heat flux of nucleate boiling in pool boiling problems, using visualisation data. Another example is the study by Kwon et al. (2020) [24], which compared the RF performance with finite volume modelling of local heat transfer coefficient across 243 cooling channels with different geometries. The study showed similar prediction accuracies between the RF and the CFD simulation using finite volume methods. Peng et al. (2020) [25] applied Convolutional Neural Network (CNN) for predicting steady-state heat conduction, resulting in accurate prediction of the temperature distribution across a random geometry in 2D space and being 3-4 folds computationally more efficient compared to numerical models offered by OpenFOAM. Another example is the application of an ANN model to predict the thermal energy transport coefficient and the thermo-hydraulic efficiency of a circular channel by Souayeh et al. (2021) [26], which resulted in 97% prediction accuracy. Moreover, Loyola-Fuentes et al. (2022) [27] applied multiple ML models to classify the two-phase flow patterns across pulsating heat pipes with an accuracy of higher than 75%, using image data.

On the other hand, hybrid and PIML modelling have received little attention in this field, and the research on these frameworks is not as widespread as one might expect compared to purely mechanistic and machine learning modelling. For instance, Jayaweera et al. (2022) [28] applied Model C in this study to predict the thermal performance of the direct contact countercurrent cooling tower, using an ANN combined with Restrep Reyes model, predicting cooling water outlet temperature. An ANN was used to predict the volumetric mass transfer coefficient to be used in the mechanistic model in an end-to-end training loop, resulting in predictions with an $R^2 = 0.99$. In another study, Loyola-Fuentes et al. (2024) [21] developed an ML modelling framework for non-dynamic heat transfer data. In that study, different machine learning-based models were used to predict the condensation heat transfer coefficient, where in one case,  dimensionless numbers were used instead of measured variables as inputs for predicting the condensation heat transfer coefficient (HTC) in microfin tubes, creating a framework similar to Model B in Figure 1. The model showed comparable results against the model with measured variables as inputs (Model A). The best model candidate of Model B in that study had a prediction error of 685 $W/m^2K$, while the pure machine learning model had a prediction error of 511 $W/m^2K$. The difference in the accuracy is only 2.8% of the average HTC of the testing set ($HTC_{ave} =$ 6,132 $W/m^2K$). The study also showed that Model B has higher accuracy extrapolating beyond the range of training data. Using a hybrid model compared to the pure machine learning one resulted in predictions with an error of 1,819 $W/m^2K$, while this metric for the pure ML model was 12,426 $W/m^2K$.

Lee et al. (2025) [29] recently applied a PIML model using XGBoost informed by the Degradation Nusselt model to predict the condensation heat transfer coefficient. The study showed that the developed PIML model is associated with stronger extrapolation capabilities compared to the pure XGBoost model. In another example, Furlong et al. (2025) [30] developed a parallel hybrid model (Model D) to predict the critical heat flux in a boiling system. However, the model training in that study did not use an end-to-end training loop, meaning that an ANN model was trained to predict the deviation between experimental critical heat flux and the predictions from a mechanistic model. Then, the pre-trained model was used on an unseen dataset to adjust the mechanistic model in seuqnce. One must note that the literature review above is by no means a comprehensive review of the ML/hybrid modelling works in heat transfer, but it provides recent examples of such modelling approaches in this field. As can be seen, the application of hybrid modelling is rather limited in this field compared to the PSE community. This study aims to further develop upon the framework developed by Loyola-Fuentes et al. (2024) [21] to accommodate choosing various hybrid modelling approaches for non-dynamic heat transfer data.

### 1.4 Model Training Algorithms

In general, while training a model, including machine learning, mechanistic or hybrid, the problem is formulated as an optimisation, to minimise the loss function ($\boldsymbol{\mathcal{L}}$) by training model parameters:

$$\boldsymbol{\vartheta}^{opt} = \arg\min_{\boldsymbol{\vartheta}} \mathcal{L}(\boldsymbol{\vartheta}) \tag{1}$$

Normally used loss functions for regression problems are $\mathcal{L}_n$ norm functions, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the most commonly used functions (equivalent to absolute error and squared error, respectively).

$$\mathcal{L} = \left|\left|e\right|\right|_n = \left[\left|\sum_i e_i\right|_n\right]^{\frac{1}{n}} \tag{2}$$

Where $e$, in a regression problem, is the error of the predictions of the target variable ($y$).

Regardless of the modelling approaches employed, in optimisation problems, the training algorithms can be categorised into two: 1) gradient-based and 2) gradient-free methods. One must note that the use of each algorithm heavily depends on the problem type, computational time constraints and differentiation properties of the model.

The majority of hybrid model studies have been employing gradient-based methods, including 1) Broyden-Fletcher-Goldfarb-Shanno (BFGS) [31–34] used by Qi et al. (1999) [35] in modelling a fixed-bed reactor, 2) Levenberg-Marquardt (LM) [36] used by Lauret, Boyer, and Gatina, (2001) [37] for modelling the Sucrose crystal growth rate, and 3) Adaptive Moment Estimation (Adam), which has gained great attention due to its efficiency in training neural networks and fast convergence in several complex problems. Moreover, it has been employed by several researchers in different applications [2,9,38,39]. Gradient-free methods have also been used in hybrid modelling, for instance, evolutionary programming in a model-based optimisation of biosurfactant production in a fed-batch culture [40] and particle swarm optimisation for modelling a sugar crystallisation process [41].

When applying gradient-based methods, three approaches are normally utilised to determine the gradients of the loss function: 1) Symbolic Differentiation (SD), 2) Numerical Differentiation (ND), and 3) Automatic Differentiation (AD). Traditionally, the first two approaches have been employed in several gradient-based optimisation problems. In many complex problems, the use of symbolic differentiation becomes rather impossible as formulating the differentiation analytically is not feasible. In problems where deriving the differentiation analytically is possible, the approach offers precise solutions. On the other hand, numerical differentiation provides an approximate solution by applying finite difference methods, which are prone to numerical errors and require high computational time [42]. The challenges associated with the first two methods led to the development of the AD method for complex and large models, addressing computational costs, implementation challenges, and accuracy of the derivative evaluations [42,43].

In automatic differentiation, the complex functions of interest are degraded into small arithmetic operations, including addition, subtraction, multiplication, division, etc., for which the derivatives are rather easy to calculate. Thereafter, a computational graph is built upon those arithmetic operations resembling the actual function of interest, as shown in Figure 2. The derivatives of the error are then calculated using the chain rule through a forward pass, followed by a back-propagation. Coupling AD with back-propagation gives a significant advantage over the ND methods as the number of parameters increases in the model [42,43]. In the past decade, several packages have been developed for this application, including Autodiff for PyTorch [44] by Facebook AI Research Lab, GradientTape in Tensorflow [45] by Google Brain Team, and more recently Grad in for JAX [46] developed by Google Brain and DeepMind. In this study, all the models are built using the Tensorflow AD package.

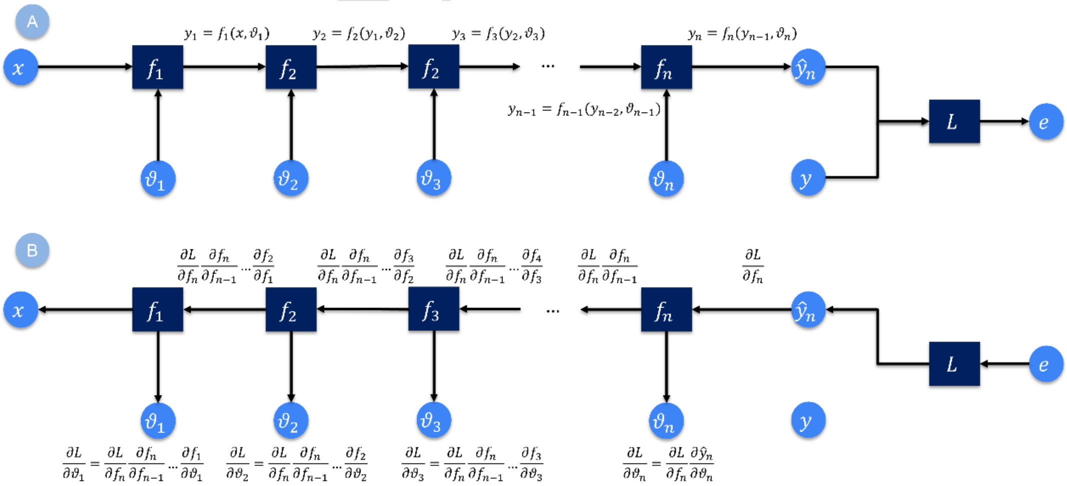
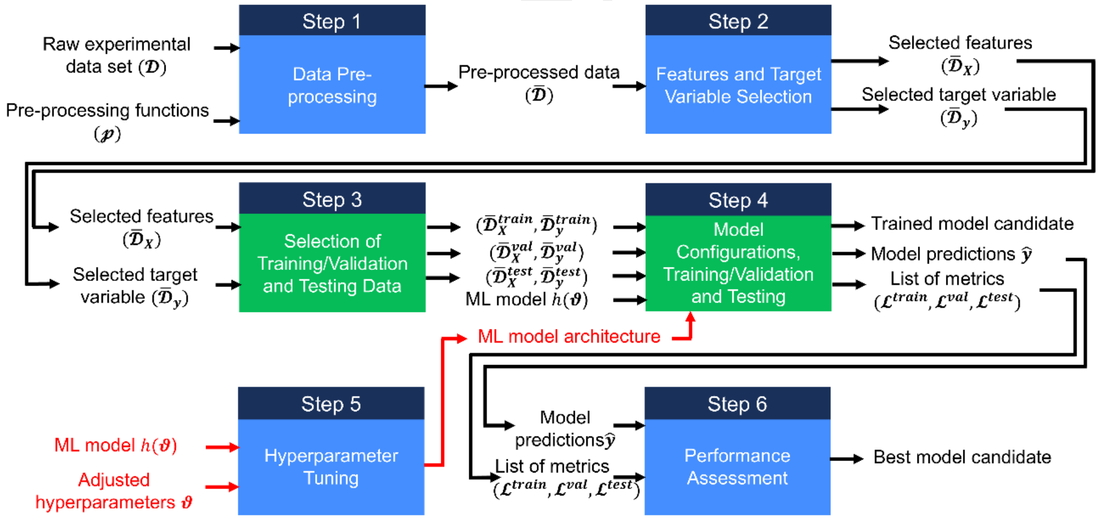

A
$y_1 = f_1(x, \vartheta_1)$
$y_2 = f_2(y_1, \vartheta_2)$
$y_3 = f_3(y_2, \vartheta_3)$
$y_n = f_n(y_{n-1}, \vartheta_n)$
$x$
$f_1$
$f_2$
$f_2$
...
$f_n$
$\hat{y}_n$
$y_{n-1} = f_{n-1}(y_{n-2}, \vartheta_{n-1})$
$L$
$e$
$\vartheta_1$
$\vartheta_2$
$\vartheta_3$
$\vartheta_n$
$y$
B
$\frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_2}{\partial f_1}$
$\frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_3}{\partial f_2}$
$\frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_4}{\partial f_3}$
$\frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}$
$\frac{\partial L}{\partial f_n}$
$x$
$f_1$
$f_2$
$f_3$
...
$f_n$
$\hat{y}_n$
$L$
$e$
$\vartheta_1$
$\vartheta_2$
$\vartheta_3$
$\vartheta_n$
$y$
$\frac{\partial L}{\partial \vartheta_1} = \frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_1}{\partial \vartheta_1}$
$\frac{\partial L}{\partial \vartheta_2} = \frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_2}{\partial \vartheta_2}$
$\frac{\partial L}{\partial \vartheta_3} = \frac{\partial L}{\partial f_n}\frac{\partial f_n}{\partial f_{n-1}}\cdots\frac{\partial f_3}{\partial \vartheta_3}$
$\frac{\partial L}{\partial \vartheta_n} = \frac{\partial L}{\partial f_n}\frac{\partial \hat{y}_n}{\partial \vartheta_n}$

**Figure 2:** Schematic representation of (A) forward pass of the model, followed by (B) back-propagation calculating the gradients of the parameters with respect to the predicted error ($e$) in a computational graph using automatic differentiation.

## 2 Systematic Hybrid Modelling Framework

The hybrid modelling framework developed in this study is founded on a framework previously developed by Loyola-Fuentes et al. (2024) [21], with changes applied in model training/testing data split and training loop, depending on the hybridisation strategy. The modelling framework allows the user to select the desired hybrid model configuration illustrated in Figure 1, agnostic to the first-principles or mechanistic model, as long as the mechanistic model is differentiable and not dynamic. The current modelling framework requires major adaptations to the data pre-processing and data splitting steps to be applicable to dynamic systems. Figure 3 represents the overview of the framework with the flow of data for each step, showing the adaptations taken from the framework developed by Loyola-Fuentes et al. (2024) [21]. Most of the steps are identical to the aforementioned framework, and the details of such steps can be found in that study. The changes applied to Step 3: Selection of Training/Validation and Testing Data provide the opportunity to split the data more effectively, leading to a more efficient training of the developed models. Moreover, changes in Step 4: Model Configuration, Training/Validation and Testing offer the user the opportunity to train different model configurations (i.e. pure machine learning or hybrid) for exploring potential model structures, predicting the desired target variables.



Step 1
Data Pre-processing
Raw experimental data set ($\mathcal{D}$)
Pre-processing functions ($\wp$)
Pre-processed data ($\bar{\mathcal{D}}$)
Step 2
Features and Target Variable Selection
Selected features ($\bar{\mathcal{D}}_X$)
Selected target variable ($\bar{\mathcal{D}}_y$)
Selected features ($\bar{\mathcal{D}}_X$)
Selected target variable ($\bar{\mathcal{D}}_y$)
Step 3
Selection of Training/Validation and Testing Data
($\bar{\mathcal{D}}_X^{train}, \bar{\mathcal{D}}_y^{train}$)
($\bar{\mathcal{D}}_X^{val}, \bar{\mathcal{D}}_y^{val}$)
($\bar{\mathcal{D}}_X^{test}, \bar{\mathcal{D}}_y^{test}$)
ML model $h(\vartheta)$
Step 4
Model Configurations, Training/Validation and Testing
Trained model candidate
Model predictions $\hat{y}$
List of metrics ($\mathcal{L}^{train}, \mathcal{L}^{val}, \mathcal{L}^{test}$)
ML model architecture
Step 5
Hyperparameter Tuning
ML model $h(\vartheta)$
Adjusted hyperparameters $\vartheta$
Model predictions$\hat{y}$
List of metrics ($\mathcal{L}^{train}, \mathcal{L}^{val}, \mathcal{L}^{test}$)
Step 6
Performance Assessment
Best model candidate

**Figure 3:** Modelling framework overview, adapted from [21], with main changes in Step 3 and Step 4

In Step 1, the data is pre-processed according to *a priori* knowledge of the system, data collection protocols, and any data analysis pipeline that the user finds suitable. It is essential at this stage to identify outliers, the range of the data for each variable that aligns with the physics of the problem, and to remove repeating datapoints from the dataset to avoid introducing biases to the model training/validation, and testing steps. A feature importance analysis shall be carried out in Step 2 to identify the most relevant variables, similar to the approach taken by Loyola-Fuentes et al. (2024) [21] (e.g. Recursive Feature Elimination with Cross-Validation (RFECV)). Steps 3 and 4 are described in the following with more details, as they are quite different from the original modelling framework:

### Step 3: Selection of Training, Validation, and Test Data

Now that the features and the target variables are identified and selected, the data shall be split in Step 3 of the workflow. The performance of the developed model highly depends on the data splitting strategy [47]. The common practice in machine learning modelling suggests splitting datasets into binary splits of training and testing sets. However, in this study, the data is split into three sets of training, validation, and test sets. During training of the model, in each iteration (epoch), the optimised model parameters are validated against a set of unseen data (validation) to assess the model's generalisation after each step. This would normally avoid overfitting and memorising patterns in the data. The following algorithm should be followed to prepare the ternary splits.

**Algorithm 3.** Data split into training, validation, and test sets:

1. Split the data first into total training (including validation) and test data using common practices in the literature. A random split of 70/30% or 80/20% is usually recommended using the Pareto principle [48]:

$$\overline{\mathcal{D}}_{total}^{train}: (\overline{\mathcal{D}}_{X,total}^{train}, \overline{\mathcal{D}}_{y,total}^{train})$$
$$\overline{\mathcal{D}}^{test}: (\overline{\mathcal{D}}_{X}^{test}, \overline{\mathcal{D}}_{y}^{test})$$

**Note:** One may choose a non-random data splitting strategy to assess the model performance, similar to the approach used in [21], where a systematic splitting method was employed to assess the extrapolability of the developed model. In this study, in addition to a random split, a similar systematic approach is used to analyse the extrapolation capabilities of the hybrid models.

2. Within the total training set obtained, split the data randomly by using 90/10%, 80/20% into training and validation sets. It is essential to split the data randomly in this step to prevent introducing biases in the training step.

$$\overline{\mathcal{D}}^{train}: (\overline{\mathcal{D}}_{X}^{train}, \overline{\mathcal{D}}_{y}^{train})$$
$$\overline{\mathcal{D}}^{val}: (\overline{\mathcal{D}}_{X}^{val}, \overline{\mathcal{D}}_{y}^{val})$$

### Step 4: Model Configuration, Training/Validation and Testing

In this step, the model configuration should be selected according to the availability of high-quality data and the mechanistic model. The user can employ different hybrid model configurations to compare their performance assessment and select the one associated with the best performance. In most applications, a deep understanding of the mechanistic model is required to decide the appropriate configuration.

Hence, it is highly recommended to apply all configurations and choose the one with more accurate predictions and generalisation capabilities.

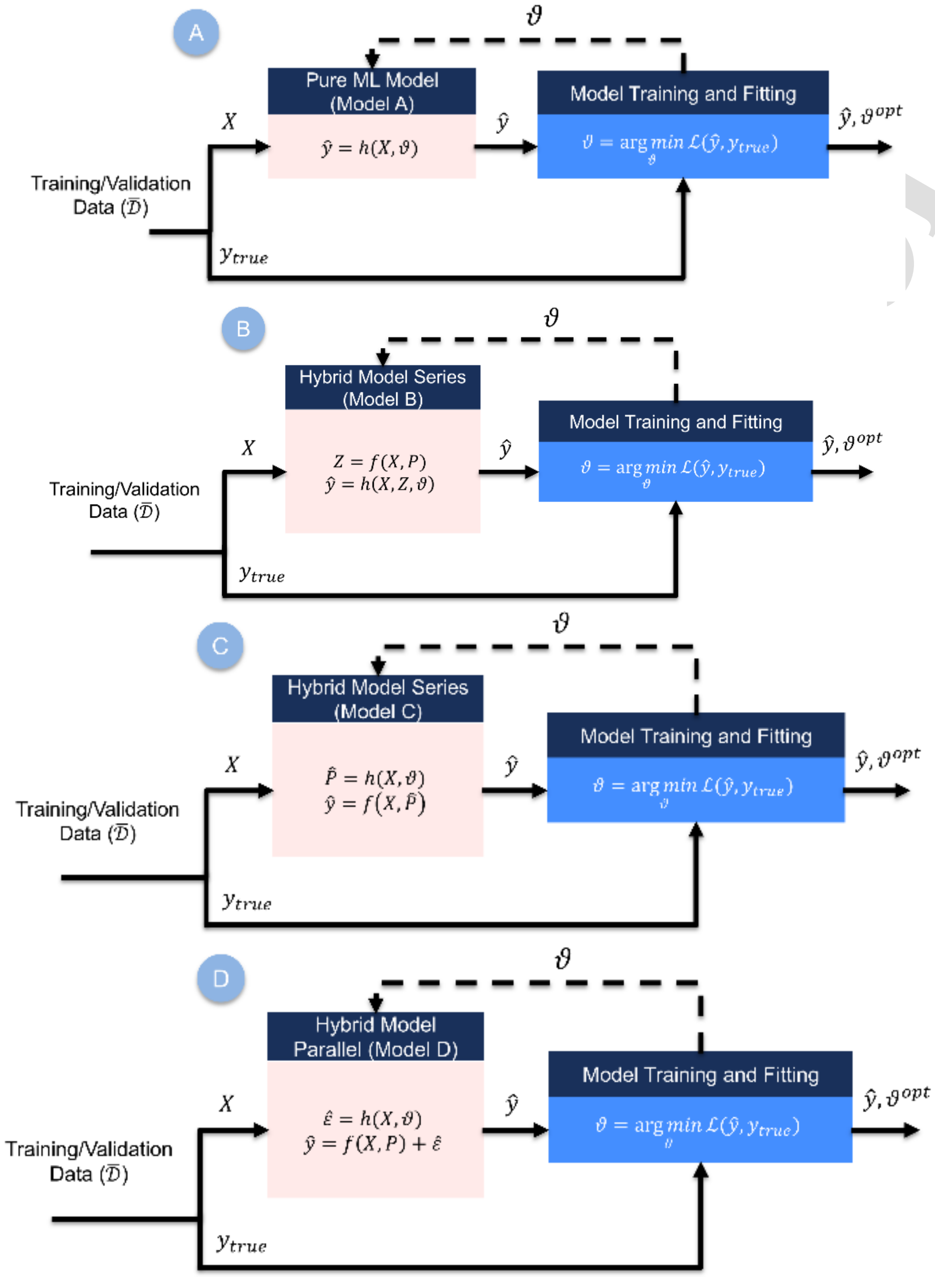

A
$\vartheta$
Pure ML Model (Model A)
$\hat{y} = h(X, \vartheta)$
Model Training and Fitting
$\vartheta = \arg\min_{\vartheta} \mathcal{L}(\hat{y}, y_{true})$
$X$
$\hat{y}$
$\hat{y}, \vartheta^{opt}$
Training/Validation Data ($\mathcal{D}$)
$y_{true}$
B
$\vartheta$
Hybrid Model Series (Model B)
$Z = f(X, P)$
$\hat{y} = h(X, Z, \vartheta)$
Model Training and Fitting
$\vartheta = \arg\min_{\vartheta} \mathcal{L}(\hat{y}, y_{true})$
$X$
$\hat{y}$
$\hat{y}, \vartheta^{opt}$
Training/Validation Data ($\mathcal{D}$)
$y_{true}$
C
$\vartheta$
Hybrid Model Series (Model C)
$\hat{P} = h(X, \vartheta)$
$\hat{y} = f(X, \hat{P})$
Model Training and Fitting
$\vartheta = \arg\min_{\vartheta} \mathcal{L}(\hat{y}, y_{true})$
$X$
$\hat{y}$
$\hat{y}, \vartheta^{opt}$
Training/Validation Data ($\mathcal{D}$)
$y_{true}$
D
$\vartheta$
Hybrid Model Parallel (Model D)
$\hat{\varepsilon} = h(X, \vartheta)$
$\hat{y} = f(X, P) + \hat{\varepsilon}$
Model Training and Fitting
$\vartheta = \arg\min_{\vartheta} \mathcal{L}(\hat{y}, y_{true})$
$X$
$\hat{y}$
$\hat{y}, \vartheta^{opt}$
Training/Validation Data ($\mathcal{D}$)
$y_{true}$

**Figure 4:** Conceptual illustration of training loop for model configurations shown in Figure 1: (A) Pure ML – Model A, (B) Hybrid Series – Model B, (C) Hybrid Series – Model C, and (D) Hybrid Model – Model D

**Algorithm 4.** Selecting model configuration, training/validation and testing:

1. Choose the desired ML model $h(X, \vartheta)$. Examples are Random Forest (RF), Artificial Neural Networks (ANN), Support Vector Machine (SVM), etc.
   Where $h$ represents the ML model, $X$ the input data, and $\vartheta$ model parameters.
2. Select the training and validation method from the following approaches, described in detail in [21]:
   a. Holdout
   b. K-fold cross-validation
   c. Ensemble
3. Select the model configuration from Figure 1, based on the following guidelines:
   a. If Model A(pure ML model) is to be used, model training is formulated as follows:
   $$\widehat{\boldsymbol{y}} = h(X, \vartheta)$$
   $$\boldsymbol{\vartheta}^{opt} = \arg\min_{\boldsymbol{\vartheta}} \boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{y}}, \boldsymbol{y_{true}})$$
   b. Else, if Model B is selected for predicting the target variable, the mechanistic model and domain knowledge are used to generate soft sensors ($\boldsymbol{Z}$) to be used instead/with actual input variables to train the ML model:
   $$Z = f(\boldsymbol{X}, \boldsymbol{P})$$
   $$\widehat{\boldsymbol{y}} = h(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\vartheta})$$
   $$\boldsymbol{\vartheta^{opt}} = \arg\min_{\boldsymbol{\vartheta}} \boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{y}}, \boldsymbol{y_{true}})$$
   c. Else, if Model C is chosen, the machine learning model is trained to predict the set of parameters ($P$) of the mechanistic model, responsible for predicting the target variable:
   $$\boldsymbol{P} = h(\boldsymbol{X}, \boldsymbol{\vartheta})$$
   $$\widehat{\boldsymbol{y}} = f(\boldsymbol{X}, \boldsymbol{P})$$
   $$\boldsymbol{\vartheta^{opt}} = \arg\min_{\boldsymbol{\vartheta}} \boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{y}}, \boldsymbol{y_{true}})$$
   d. Else, if Model D, predict the target variable with the mechanistic model, while training the ML model to predict an additive correction term ($\varepsilon$) to adjust the predictions against the actual measured target variables:
   $$\boldsymbol{\varepsilon} = h(\boldsymbol{X}, \boldsymbol{\vartheta})$$
   $$\widehat{\boldsymbol{y}} = f(\boldsymbol{X}, \boldsymbol{P}) + \boldsymbol{\varepsilon}$$
   $$\boldsymbol{\vartheta^{opt}} = \arg\min_{\boldsymbol{\vartheta}} \boldsymbol{\mathcal{L}}(\widehat{\boldsymbol{y}}, \boldsymbol{y_{true}})$$
   Where $h$ represents the ML model with $\boldsymbol{\vartheta}$ as its hyperparameters, while $f$ is the mechanistic model with $\boldsymbol{P}$ as its parameters.

4. Carry out a prediction on the test dataset for the trained models in the previous steps for performance assessment in Step 6.

This study focuses on a hybrid modelling framework that integrates mechanistic and machine learning model components for predicting the desired target variable, using various model configurations. To compare the results of the models, the same neural network architecture (*i.e.*, number of hidden layers, number of neurons per hidden layer, and hidden layer activation functions) can be used to provide a fair basis of comparison across different configurations, or the hyperparameters of each neural network can be tuned in every model configuration and select the best model across the tuned ones. In this case, the first option was chosen, as the purpose of this study is to highlight the importance of the model configuration (interactions between first principles and machine learning models), rather than the use of the best possible ML model architecture. The changes in other hyperparameters, such as the number of neurons in the outlet layer, depend on the model configuration itself, which further leads to changes in the output layer activation function. Finally, different learning rates are needed because of the different model configurations and their corresponding vanishing/exploding gradient rates during the training stage. As stated in the preceding study of the current paper, Step 5 is rather optional to tune the hyperparameters of the developed model, but it helps to avoid unnecessarily complex models for the system under study. Thereafter, in Step 6, the developed models in Step 3 are compared against each other in terms of their prediction and learning curves using the error metrics such as Mean Absolute Error (*MAE*), Mean Squared Error (*MSE*), and other similar metrics. In addition to these metrics, the percentage of points outside of a fixed threshold (e.g., 20%) of the experimental values ($\hat{\boldsymbol{y}}$). More details of the framework can be found in the study by Loyola-Fuentes et al. (2024) [21].

The main advantage of the framework developed in this study over other existing models in the literature lies in the integration of the state-of-the-art ML and differentiable mechanistic models for modelling non-dynamic heat transfer data, agnostic to the system. The framework provides the possibility of training various model configurations, and the user is required to select the model that explains the variation of the data the most. Moreover, the framework is designed as such to provide the opportunity for an end-to-end training of the hybrid models, which is more efficient compared to using a pre-trained ML model.

## 3 Application Example: Condensation Heat Transfer Coefficient in Microfin Tubes

### 3.1 Background in Microfin Tubes

Over the past four decades, the utilisation of enhanced surfaces to boost condensation heat transfer within horizontal channels has become standard practice. Consequently, microfin tubes, as an engineering innovation that enhances thermal exchange, are extensively employed in heat transfer devices for various HVAC and cooling systems. Indeed, since their conception by Fujie et al. (1975) [20] microgrooved tubes have garnered considerable interest due to their capacity to deliver substantial improvements in heat transfer (80-180%) with only a modest rise in pressure loss (20-80%) when compared to a comparable smooth tube under identical operating parameters. They achieve enhanced thermal exchange through (i) expanding the effective heat transfer surface; (ii) generating heightened turbulence within the liquid layer; and (iii) leveraging surface tension phenomena to promote condensate removal [20]. Common microfin tubes are characterised by fin tip diameter ($d_f$), number of fins ($n_f$), helix angle ($\beta$), apex angle ($\alpha$) and fin height ($h_f$).

These characteristics are illustrated in Figure 5, which presents a fundamental configuration of a microfin tube. Typically, they have an internal diameter ranging from 3 to 15 mm, featuring a single array of 40-70 fins with a helical inclination ($\beta$) varying from 0° to 30°, fin depth ($h_f$) between 0.1 and 0.25 mm, and triangular or trapezoidal fin profiles with an apex angle (γ) spanning from 25° to 90°. The heat transfer coefficient during condensation within microfin tubes is dependent on the complex interactions among the tube's parameters (e.g., geometry, operational parameters, flow regime, etc.) and heat transfer mechanisms (e.g., forced convection, temperature gradients, etc.).

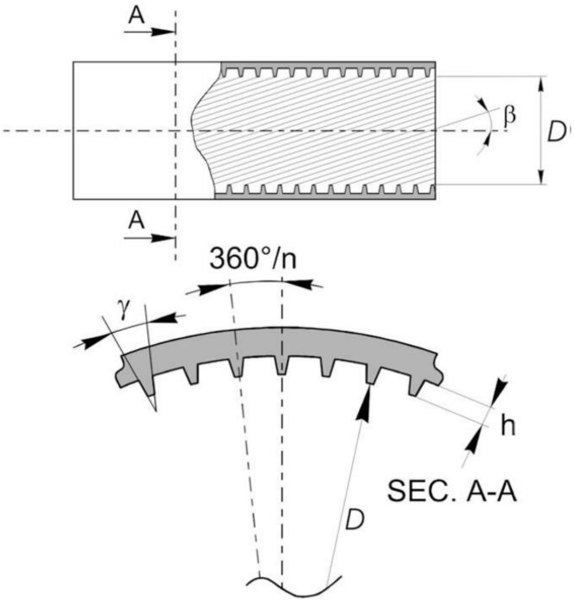


**Figure 5:** Basic design of a microfin tube reprinted with permission from [20]. © 2009 Elsevier.

Although the Cavallini et al. (2009) [20] model offers good predictive capabilities, but the new requirements of the refrigeration and air conditioning sector in terms of controls and energy cost minimisation call for a novel class of models that could increase the accuracy of prediction while maintaining the reliability and stability of traditional ones. Loyola-Fuentes et al. (2024) [21] explored the main capabilities of machine learning tools in estimating condensation heat transfer coefficients, showing their limitations in terms of predictability. This work aims to take a step forward by attempting to demonstrate that the hybridisation of machine learning tools can improve the overall accuracy and predictability of mechanistic models, enabling advanced features in the design of smart heat exchangers.

### 3.2 Mechanistic Model

The Cavallini et al. (2009) [20] model was developed by regressing the empirical constants on 558 experimental data points selected among more than 4000 available in the whole database. These experimental datapoints were those presenting low experimental uncertainty, available temperature difference, different tube geometries and operative fluids. The selected data points were: 1) 186 datapoints by Cavallini et al. [49–51] for R22 and R407C inside a horizontal microfin tube and for R134a and R410A, 2) 96 points by Haraguchi (1994) [52] for R134a and R123, 3) 106 datapoints by Kedzierski and Gonclaves (1997) [53] for R32, 4) 52 datapoints by Miyara et al. (2002) [54] for R410A, 5) 39 by Kim et al. (2002) [54] for R22 and R410A, 6) 20 by Zilly et al. (2003) [55] for $CO_2$, 7) 32 by Colombo et al. (2006) [56] for R134a, and 8) 17 by Uchida et al. (1997) for R22 [57].

The condensation heat transfer coefficient is defined with reference to the heat transfer area of the smooth tube with tube diameter ($D$) equal to the fin tip diameter of the microfin tube according to the following equation:

$$HTC = \frac{q}{\pi D L}\frac{1}{\Delta T} \tag{3}$$

The condensation HTC is estimated by combining the heat transfer coefficient for the $\Delta T$ independent zone ($HTC_A$) and the one for $\Delta T$ dependent zone ($HTC_D$) using the following equation:

$$HTC = [HTC_A^3 + HTC_D^3]^{\frac{1}{3}} \tag{4}$$

The first term in Eq. (4) corresponds to the forced convective heat transfer coefficient and can be determined as the product of the HTC for a smooth tube ($HTC_{AS}$) by a function ($A$) of the geometry enhancement factor ($R_x$) and the Froude number ($Fr$) [58] using Eqs. (5)-(11). The term $C$ acts to lower the heat transfer coefficient when the fin number ($n_g$) is greater than the optimal value ($n_{opt}$) for the given diameter.

$$HTC_A = A \cdot C \cdot HTC_{AS} \tag{5}$$

$$HTC_{AS} = HTC_{LO}\left[1 + 1.128x^{0.817}\left(\frac{\rho_L}{\rho_V}\right)^{0.3685}\left(\frac{\mu_L}{\mu_V}\right)^{0.2363}\left(1 - \frac{\mu_V}{\mu_L}\right)Pr_L^{-0.1}\right] \tag{6}$$

$$HTC_{LO} = 0.023\frac{\lambda_L}{d_f}\left(\frac{Gd_f}{\mu_L}\right)^{0.8}Pr_L^{0.4} \tag{7}$$

$$A = 1 + aFr^b(R_x - 1)^c \tag{8}$$

$$Fr = \frac{G^2}{gd_f(\rho_L - \rho_V)^2} \tag{9}$$

$$R_x = \frac{\left[\dfrac{2h \times n_g\left(1 - \sin\left(\dfrac{\gamma}{2}\right)\right)}{\pi d_f \cos\left(\dfrac{\gamma}{2}\right)} + 1\right]}{\cos(\beta)} \tag{10}$$

$$\begin{aligned} &C = 1 \text{ if } \left(\frac{n_{opt}}{n_g}\right) \geq 0.8 \\ &C = \left(\frac{n_{opt}}{n_g}\right)^d \text{ if } \left(\frac{n_{opt}}{n_g}\right) < 0.8 \\ &n_{opt} = ed_f + f \end{aligned} \tag{11}$$

The HTC of the $\Delta T$ dependent zone ($HTC_D$) can be determined from Eqs. (12-(17(17) by using the parameter $C$ in Eq. (11), the geometry enhancement factor ($R_x$) and calculating the HTC of the $\Delta T$ dependent zone for smooth tube ($HTC_{DS}$) [58].

$$HTC_D = C[gx^h(R_x - 1)^k C_1^m + 1]HTC_{DS} + C(1 - x^n)R_x HTC_{LO} \tag{12}$$

$$HTC_{DS} = \frac{0.725}{1 + 0.741\left(\dfrac{1-x}{x}\right)^{0.3321}}\left[\frac{\lambda_L^3 \rho_L(\rho_L - \rho_V)gh_{LV}}{\mu_L d_f \Delta T}\right]^{0.25} \tag{13}$$

$$\begin{aligned} &C_1 = 1 \text{ if } J_V \geq J_V^* \\ &C_1 = J_V/J_V^* \text{ if } J_V < J_V^* \end{aligned} \tag{14}$$

$$J_V = \frac{xG}{[gd_f\rho_V(\rho_L - \rho_V)]^{0.5}} \tag{15}$$

$$J_V^* = 0.6\left[\left(\frac{7.5}{4.3X_{tt}^{1.111} + 1}\right)^{-3} + 2.5^{-3}\right]^{-\frac{1}{3}} \tag{16}$$

$$X_{tt} = \left(\frac{\mu_L}{\mu_V}\right)^{0.1}\left(\frac{\rho_V}{\rho_L}\right)^{0.5}\left(\frac{1-x}{x}\right)^{0.9} \tag{17}$$

Where $J_V$ is the dimensionless vapour velocity, $J_V^*$ represents the transition vapour velocity, and $X_{tt}$ is the Martinelli parameter. The empirical parameters in Eqs. (8), (11), and (12) have to be estimated by training the model over the collected experimental dataset. The model is valid for tubes with helical fins having a fin height to diameter ratio $h_f/D$ less than 0.04. This model should be applied to halogenated refrigerants and carbon dioxide with reduced pressure $0.1 < P_{red} < 0.67$, vapour quality $0 < x < 1$ and mass velocity ranging between $90 < G < 900$ kg m$^{-2}$ s$^{-1}$.

### 3.3 Dataset

The data set for condensation in microfin tubes used in this paper is a larger data set in addition to the data used by Loyola-Fuentes et al. 2024 [21] consisting of 4,122 data points used in the preceding studies [20] (Batch 1) and an additional 1,586 datapoints (Batch 2) from other studies in this area [59–72], resulting in 5,708 datapoints in total. The added experimental data points include: new refrigerants, among those: low GWP fluids R1234yf and R1234ze, R513, R1123/R32 mixture, additional microfin tubes with smaller diameters, from 2.17 mm up to 5.21, and add more data for traditional Hydrofluorocarbon (HFC) fluids. Table 3 shows the experimental input variables and their ranges.

**Table 3:** Experimental input data to the models

| Variable | Symbol | Units | Range | Variable | Symbol | Units | Range |
|---|---|---|---|---|---|---|---|
| **Vapour quality** | $x$ | $-$ | [0.02 - 1] | **Heat of vaporisation** | $h_{LV}$ | $J\ kg^{-1}$ | [8.5E4 – 2.93E5] |
| **Refrigerant mass flux** | $G$ | $kg\ m^{-2}s^{-1}$ | [54.97 – 1529.46] | **Reduced pressure** | $P_{red}$ | $-$ | [0.01 – 0.63] |
| **Liquid density** | $\rho_L$ | $kg\ m^{-3}$ | [867.26 – 1480.84] | **Surface temperature** | $T_s$ | $°C$ | [-25.0 – 74.11] |
| **Liquid thermal cond.** | $\lambda_L$ | $W\ m^{-1}K^{-1}$ | [0.05 – 0.14] | **Surface to wall temp. difference** | $T_s - T_w$ | $°C$ | [0.23 – 18.07] |
| **Liquid heat capacity** | $Cp_L$ | $J\ kg^{-1}K^{-1}$ | [881.02 – 2280.9] | **Fin tip diameter** | $d_f$ | $m$ | [1.98E-3 – 1.6E-2] |
| **Liquid viscosity** | $\mu_L$ | $Pa\ s$ | [8.18E-5 – 4.33E-4] | **Fin height** | $h_f$ | $m$ | [1.00E-4 – 6.35E-4] |
| **Liquid surface tension** | $\sigma_L$ | $N\ m^{-1}$ | [1.45E-3 – 1.7E-2] | **Helix angle** | $\beta$ | ° | [0 – 40.0] |
| **Vapour density** | $\rho_V$ | $kg\ m^{-3}$ | [6.99 – 166.95] | **Apex angle** | $\gamma$ | ° | [0 – 90.0] |
| **Vapour thermal cond.** | $\lambda_V$ | $W\ m^{-1}K^{-1}$ | [8.6E-3 – 2.6E-2] | **Number of fins per tube** | $n_f$ | $-$ | [10 - 82] |
| **Vapour heat capacity** | $Cp_V$ | $J\ kg^{-1}K^{-1}$ | [615.61 – 2587.86] | **Tube length** | $L_f$ | $m$ | [0.15 – 6.4] |
| **Vapour viscosity** | $\mu_V$ | $Pa\ s$ | [1.03E-5 – 1.81E-5] | | | | |

## 3.4 Approach

In this study, a thorough approach is used to analyse the performance of the various modelling configurations introduced in two different scenarios and subsequent sub-scenarios, listed in Table 4: 1) Scenario I: Testing interpolation capabilities, and 2) Scenario II: Testing extrapolation capabilities. Scenario I allows for splitting the data randomly using common practices of ML models using the Pareto Principle, as suggested in Step 3 of the framework. By randomly splitting data into training/validation and testing sets, the test set will contain datapoints that are within the range of the training data. However, in Scenario II, the data is split systematically by considering a certain range of mass flux ($G$) in the training and validation, while holding out the rest for the test set. Details of such a splitting strategy can be found in Table 5. Mass flux is selected as the variable for splitting the data since 1) it is usually a manipulated variable in experimental analysis, and 2) it is an important feature based on the RFECV results, shown later in this study. In this scenario, the extrapolation capabilities of the developed models are evaluated. Moreover, in each scenario, different model configurations, shown in Figure 1, are developed to predict the condensation HTC in microfin tubes.

**Table 4:** Summary of the scenarios considered for the application example

| Scenario | Description |
|---|---|
| **IA** | Random data split, pure machine learning (Model A) |
| **IC** | Random data split, hybrid-series model (Model C) |
| **ID** | Random data split, hybrid-parallel model (Model D) |
| **IE** | Random data split, pure mechanistic model (Model E) |
| **IIA** | Systematic data split, pure machine learning (Model A) |
| **IIC** | Systematic data split, hybrid-series model (Model C) |
| **IID** | Systematic data split, hybrid-parallel model (Model D) |
| **IIE** | Systematic data split, pure mechanistic model (Model E) |

Table 4 provides a summary of the scenarios analysed in this study, with their description, while it should be mentioned that for a pure mechanistic approach (Model E), the parameters of the model are estimated over the dataset under study, using the original parameters as the initial guess. The trained mechanistic model is then used in parallel with the ML model to configure Model D (hybrid-parallel). In Model C, the ML component assists in estimating the mechanistic model parameters within an integrated training loop.

**Table 5:** Scenario I and II data splitting strategy and details

| Scenario | Training/Validation Split [%] | Testing Split [%] | Strategy | Variable | Training/Validation Range |
|---|---|---|---|---|---|
| **I** | 90/10 | 20 | Random | - | - |
| **II** | 90/10 | 18 | Systematic | $G$ | $G \in [50, 600] kgm^{-2}s^{-1}$ |

## 4 Model E – Mechanistic Model

In this section, the parameters in the mechanistic model described above are optimally estimated using the new dataset to have a fair basis of comparison across all proposed models. Moreover, as discussed in the previous section, Model E provides the mechanistic component in Model D, which uses the correction term to adjust the prediction from the mechanistic model, aiming for a more accurate prediction. To estimate the parameters of the mechanistic model, the built-in TensorFlow's optimisation algorithm is employed and is carried out in a computer with Intel® Core™ Ultra 7 155H, 1400 MHz CPU using Python.

Table 6 provides the details of the optimisation algorithm utilised for the estimation of parameters of the mechanistic model, which are later used in scenarios IE and IIE. In both scenarios, the Adam optimisation algorithm is employed with 0.01 as the learning rate. For training the model, a batch size equivalent of 10 datapoints is selected as the sample size in each iteration (epoch) for a total number of 200 epochs (i.e. iterations).

**Table 6:** Parameter estimation settings for the mechanistic model

| Parameter | Values |
|---|---|
| **Batch size** | 10 |
| **Epochs** | 200 |
| **Optimisation Algorithm** | Adam |
| **Learning Rate** | 0.01 |
| **Number of Parameters** | 11 |

Table 7 shows that for both scenarios IE and IIE, new parameters have been estimated that are, in most cases, aligned with the ones in the model proposed by Cavallini et al. (2009) [20]. Since the original parameters are used as an initial guess, it helped the algorithm to optimise the parameters in a reasonable time, resulting in 102 s for Scenario IE and 144 s for Scenario IIE. More details of the results are provided in the Performance Assessment section, together with the machine learning and hybrid model configurations.

**Table 7:** Parameters of the mechanistic model in 1) Cavallini et al. (2009) [20], 2) Scenario IE, and 3) Scenario IIE.

| Model | a | b | c | d | e | f | g | h | k | m | n |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Cavallini et al. (2009)** [20] | 1.119 | -0.3821 | 0.3586 | 1.904 | 4064.4 | 23.257 | 2.4 | 0.1206 | 1.466 | 0.6875 | 0.087 |
| **IE** | 1.175 | -0.2228 | 0.5267 | 3.08 | 4064.22 | 28.287 | 2.4072 | 0.3791 | 0.8256 | 0.68 | 0.0869 |
| **IIE** | 1.0709 | -0.3029 | 0.3236 | 2.9839 | 4064.54 | 28.404 | 2.52 | 0.349 | 0.999 | 0.68 | 0.0869 |

## 5 Application of the Regression Framework to the Case Study

In this section, a step-by-step application of the framework, detailed in Section 2 is followed by the results of the model predictions and the assessment of their performance. It must be noted that, in addition to the ML and hybrid ML models, the mechanistic model performance is also analysed and compared to the other model configurations.

### 5.1 Data Pre-processing

As described in the framework, the data set must be screened, and proper filtering functions have to be employed to prepare the data set for modelling purposes, including outliers, data violating the physics of the problem, etc. In this case study, the only filtering procedure taken is to keep the data corresponding to an HTC of less than 24,000 $W/m^2K$, to maintain consistency with the results of Cavallini et al. (2009) [20]. As a result of this filtering procedure, the number of datapoints reduces from 5,708 to 5,646, equivalent to 99% of the total data. According to the quality of the collected data, no further filtering function was used.

### 5.2 Features and Target Variables Selection

Across all models, the same input variables described in Table 3 are used to predict the condensation HTC in both scenarios. Following the selection of input and target variables, a feature importance analysis is carried out using recursive feature elimination (RFECV) with 5-fold cross-validation. This is done via the Random Forest regressor from the Scikit-learn library in Python. Figure 6 demonstrates the results of such analysis over the filtered data, showing vapour quality $(x)$, mass flux $(G)$, and fin tip diameter $(d_f)$ as the most important variables with approximately 0.4, 0.22, and 0.14 as their score, respectively, followed by liquid thermal conductivity $(\lambda_L)$, wall temperature $(T_S - T_W)$, and the remaining input variables with an importance of smaller than 0.05, showing that they are irrelevant for the prediction of HTC. The results of the dataset used in this study are rather comparable with the study of Loyola-Fuentes et al. (2024) [21], in which the first two important variables remain the same. However, the third important variable in that study was liquid thermal conductivity, and the fin tip diameter appeared as a rather irrelevant variable. It is expected that the liquid and vapour-related physical properties have different importance scores. For instance, the thermal conductivity of the liquid phase is far more important than the vapour phase. This is expected and can be explained as in film-wise condensation of a vapour phase on a cold surface, the liquid phase imposes a larger thermal resistance compared to the vapour phase. This is also seen in the mechanistic model used in this study, as $\lambda_V$ is not used in the determination of the condensation heat transfer coefficient, while $\lambda_L$ is used in Equations (7) and (13). Moreover, considering the fin tip diameter, the results confirm what had already been proposed by Cavallini et al (2009) [20] when developing the mechanistic model. In fact, it is well known that the heat transfer performance increases as the inner diameter of the pipe decreases.

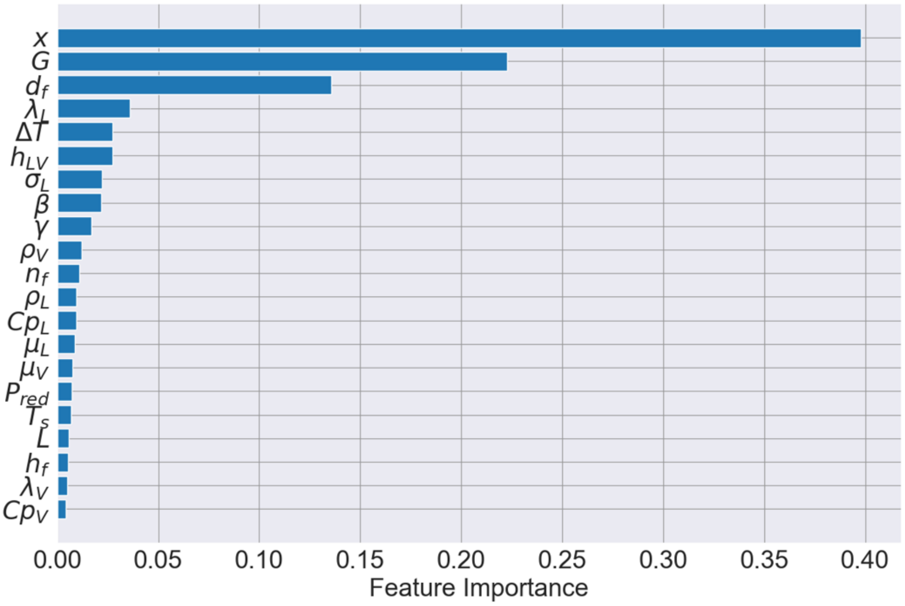


**Figure 6:** Feature importance analysis on the input variables for the prediction of HTC.

To have a deeper understanding of the results in comparison with the study of Loyola-Fuentes et al. (2024) [21], the distribution of the most important variables for the prediction of HTC (i.e., $x$, $G$, $d_f$) are analysed for the two batches of data: 1) Batch 1, including data from [21] and 2) Batch 2: new data added in this study. Figure 7 illustrates the distribution of the three most important variables across the two batches of data, showing rather similar vapour quality across the two batches, which is expected as the experiments shall cover the entire range of the variable between 0 and 1, ensuring that several possible condensation mechanisms and flow characteristics are covered. For mass flux, in Batch 2, a wider distribution is observed, while having an average $G \approx 350\,kg/m^2s$ in Batch 1 and $G \approx 400\,kg/m^2s$ in Batch 2. The major change lies in the fin tip diameter, as Batch 2 contains a distribution of fin tip diameters with an average of $d_f \approx 0.0035\,m$, while this variable shows a rather narrow distribution in Batch 1 with $d_f \approx 0.0085\,m$ as the average value. It is reasonable to assume that, due to the very narrow distribution in Batch 1, this variable was not identified as relevant in [21] using the RFECV method, while it is determined as the third important variable for the prediction of HTC in the combined dataset used in this study. This shows the importance of having a wide range of data per feature when dealing with data-driven models.

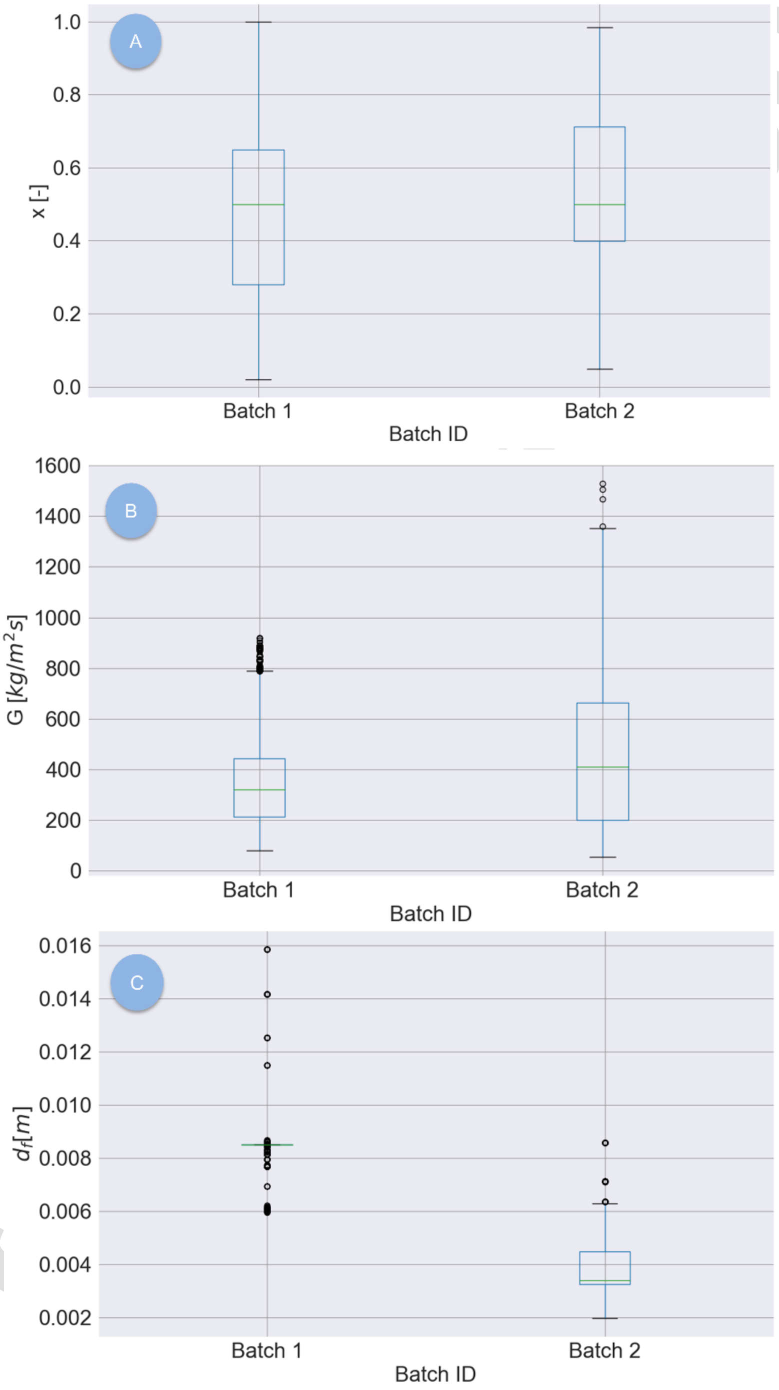


**Figure 7:** Distribution of A) Vapour quality ($x$), B) Mass flux ($G$), and C) Fin tip diameter ($d_f$) for Batch 1 and Batch 2 of the condensation in microfin tubes data.

### 5.3 Selection of Training, Validation and Testing Data

Considering the two scenarios discussed above, two different splitting strategies have been taken into account to assess both interpolation and extrapolation capabilities of the developed models. In Scenario I, as shown in Table 5, the dataset is randomly split into a total training (including training and validation) and testing sets by 80/20%, while the total training data is split by 90/10% into training and validation sets afterwards. However, in Scenario II, the data is split into total training and testing sets by having a similar ratio to the first scenario (82/18%), using $G$ as the dividing variable, comprising the total training data with $50 \leq G \leq 600\ kg/m^2s$ and $G > 600\ kg/m^2s$ as the test set, see Figure 8 for visual illustration of the data split. Thereafter, the total training data is randomly split by a ratio of 90/10% into training and validation sets.

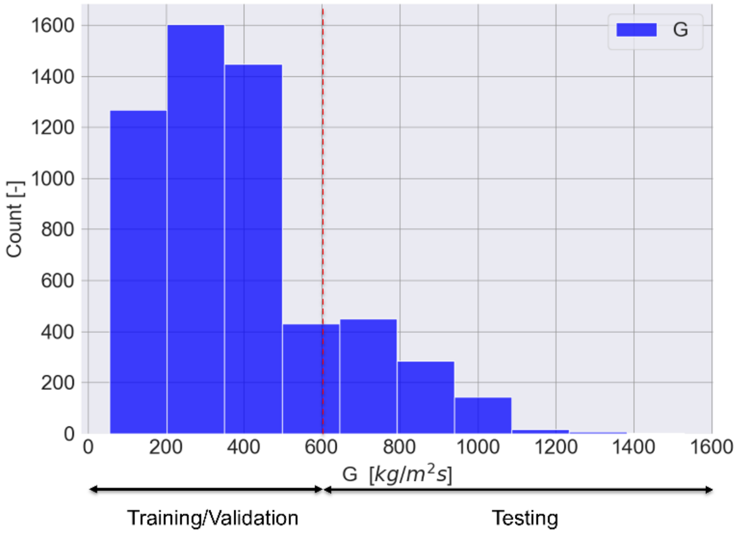


**Figure 8:** Distribution of mass flux ($G$) considering the data split for Scenario II

### 5.4 Model Configuration, Training/Validation, and Testing

At this stage, the models are trained, validated and tested for the scenarios listed in Table 4, using the training loops described in Step 4 of the modelling framework. The machine learning selected for this application is an Artificial Neural Network (ANN) with the hyperparameters listed in Table 8. The ANN models all have two hidden layers, apart from the input and the output layer, each having 32 neurons with ReLU as the activation function. The output layer activation function needs to be decided according to the model configuration. It must be noted that only the input features are normalised for the training/validation stage, and the target variable is predicted without scaling.

**Table 8:** List of hyperparameters for the ANN model, shared in all model configurations

| Hyperparameter | Values |
|---|---|
| **Number of hidden layers** | 2 |
| **Number of neurons per hidden layer** | [32, 32] |
| **Activation function per hidden layer** | [ReLU, ReLU] |
| **Batch size** | 10 |
| **Epochs** | 200 |
| **Loss** | MAE |
| **Optimiser** | Adam |

- **Scenario I**

In Scenario I (random data splitting strategy), the models are trained and validated according to the training loops described in the framework and illustrated in Figure 4. The output layer activation function and learning rate depend on the selected model configuration and the inherent nature of the neural network outputs. In other words, the behaviour of the activation function shall be consistent with the predicted variable by the neural network. In case of vanishing/exploding gradients, learning rate and the output layer activation function can be adjusted. For model C, these are adjusted to avoid exploding and vanishing gradients in the training stage and for Model A and Model D, the output layer activation function to ensure that the output of the ANN model fits the purpose of prediction. For instance, the output of the ANN in model D is an additive error ($\varepsilon$), which can take negative or positive values. Hence, a linear activation function is selected for the output layer that allows for both values as the output of the ANN model. For Model C, 0.001 is used as the learning rate and Softplus as the output layer activation function to avoid vanishing gradients in the training loop. The learning rate for the other models (A, C, and D) is 0.01, as this relatively high rate showed a reasonable learning curve for these models. An important difference among the models is the number of parameters, as this depends on the architecture of the ANN component for Models A, C, and D. All models share the same architecture in terms of the number of hidden layers and their neurons. However, the output layer is different for Model C, as its output layer has 11 neurons (equal to the number of parameters of the mechanistic model), while Models A and D have a single output for predicting the HTC and $\varepsilon$, respectively. In doing so, the total number of parameters for the models is 1,921 for Model A and D, 2,251 for Model C, and 11 for the mechanistic model. The summary of the parameters described above is provided in Table 9.

**Table 9:** Hyperparameters, computational time, number of parameters, and losses for models in Scenario I.

| | Model A | Model C | Model D | Model E |
|---|---|---|---|---|
| **Training Computational Time [s]** | 135 | 180 | 130 | 102 |
| **Number of parameters** | 1,921 | 2,251 | 1,921 | 11 |
| **Learning rate** | 0.01 | 0.001 | 0.01 | 0.01 |
| **Last layer activation function** | ReLU | Softplus | Linear | - |
| **Training loss $[W/m^2K]$** | 559 | 400 | 504 | 1,297 |
| **Validation loss $[W/m^2K]$** | 719 | 565 | 659 | 1,414 |

Figure 9 illustrates the training and validation curves for all models in this scenario. The learning curve for model E is different from the other models, as the mechanistic model is able to learn very fast at the initial epochs (up to epoch 25), but the model does not learn much afterwards, as both training and validation losses flatten around 1,297 and 1,414 $W/m^2K$, respectively. This suggests that the original parameters as the initial guess for Model E are rather close to the optimised values. Details can be found in Table 7. On the other hand, the other models' learning curves provide evidence that the training has been more effective. Model A's learning curve is quite steep, with an initial loss value starting at 6,000 $W/m^2K$ and reaching 1,000 $W/m^2K$ already in the first 20 iterations, showing that the pure ML model is quite far from a trained model when starting the training loop, and it reaches 559 and 719 $W/m^2K$ by the end of training for the training and validation sets, respectively. However, it can be clearly seen that the hybrid models (Models C and D) start from much lower loss values compared to Model A. This shows that the mechanistic model integrated with the ANN assists the model to initialise more reasonably, while facilitating more efficient training and validation of the models. Model D final training and validation losses are 504 and 659 $W/m^2K$, slightly more accurate than Model A. The final training and validation losses of Model C are 400 and 565 $W/m^2K$ respectively, the lowest among all trained models.

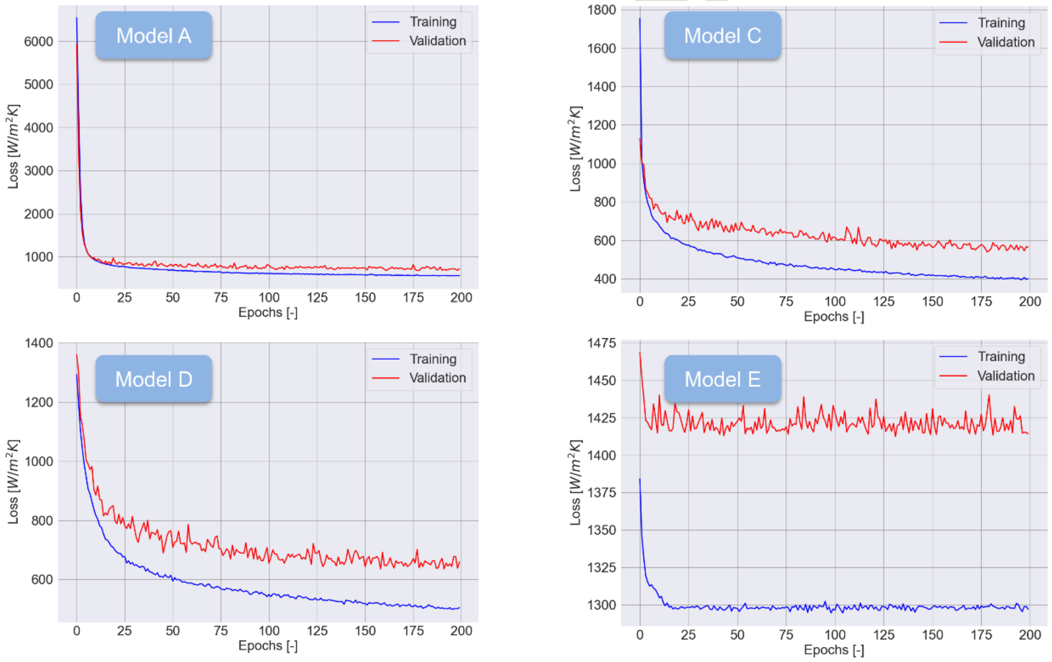


**Figure 9:** Training and validation curves of Scenario I for models A) Pure ML model, C) Hybrid – series, D) Hybrid – parallel, and E) Pure mechanistic.

Analysing the training/validation curve of Model D, reveals that the training curve has not yet reached a plateau after 200 epochs, which may indicate that the training has not been sufficient, and the model may require more iterations to be fully trained. However, the validation curve shows otherwise, as it has reached a plateau. This means that by further training the model, the two curves deviate further from each other, and it can be an indication of overfitting. Hence, 200 epochs appear to have been sufficient for this application. However, larger number of epochs can be explored if found necessary.

- **Scenario II**

In the second scenario, the same model architecture as in Scenario I has been implemented. Similarly, Model C in this scenario has a lower learning rate compared to the other models and employs Softplus as the output activation function to avoid vanishing gradients in the training loop. Table 9 shows these hyperparameters together with the computational time required for training, and the corresponding training and validation losses.

**Table 10:** Hyperparameters, computational time, number of parameters, and losses for models in Scenario II.

| | **Model A** | **Model C** | **Model D** | **Model E** |
|---|---|---|---|---|
| **Computational Time [s]** | 141 | 150 | 112 | 144 |
| **Number of parameters** | 1,921 | 2,251 | 1,921 | 11 |
| **Learning rate** | 0.01 | 0.001 | 0.01 | 0.01 |
| **Last layer activation function** | ReLU | Softplus | Linear | - |
| **Training loss [$W/m^2K$]** | 518 | 359 | 465 | 1301 |
| **Validation loss [$W/m^2K$]** | 555 | 467 | 517 | 1288 |

The learning curves illustrated in Figure 10 are relatively similar to the one presented for Scenario I. However, in Model E it can be seen that the validation loss is approximately 1,288 $W/m^2K$, while the training loss is slightly higher 1,301 $W/m^2K$. Although this difference is not significant, it shows that the mechanistic model is prone to underfitting. On the contrary, the other models' behaviour is rather similar to their counterparts in Scenario I. This behaviour would indicate that a similar performance can be expected from all models in both scenarios. Nevertheless, the training and validation data across the two scenarios are different. To draw a proper conclusion on the models' performance in both scenarios, it is necessary to assess the model prediction on a completely unseen data set (i.e. testing set).

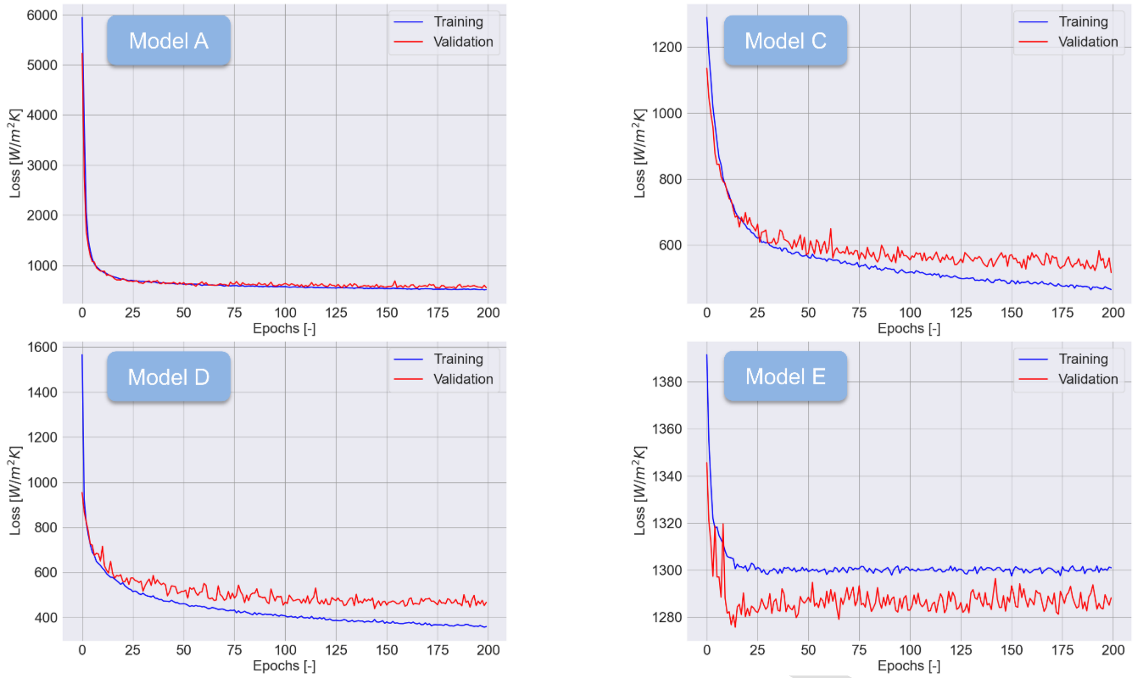


**Figure 10:** Training and validation curves of Scenario II for models A) Pure ML model, C) Hybrid – series, D) Hybrid – parallel, and E) Pure mechanistic.

## 5.5 Performance Assessment

The performance of the developed models is assessed by analysing the prediction errors over training, validation, and test data sets together with their out-of-range data points for both scenarios. These values are reported for pure ML, hybrid, and pure mechanistic models. The evaluation of the results is carried out separately for each scenario in the following.

- **Scenario I**

The prediction error of the models in this scenario on the test set, presented in Table 11, clearly indicates that the ML model and both hybrid configurations outperform the mechanistic model by having approximately 55% more accurate predictions and significantly reducing the out-of-range datapoints by almost 28%. The main reason behind this improvement is the strong interpolation capabilities of the ML component used in models A, C, and D. Figure 11 illustrates the parity plot of the models in this scenario for the test set, in which the superiority of these models over Model E, the mechanistic model. It must be noted that a similar observation was made by Loyola-Fuentes et al. (2024) [21], that showed the pure ML model with grid search has approximately 50% more accurate predictions, while the number of data points out of a 20% error range was reduced by 25%.

**Table 11:** Model performance results for Scenario I, including MAE and out-of-range data

| Model | MAE Training (Validation) $[W/m^2K]$ | %Out-of-Range Training (Validation) | MAE Testing $[W/m^2K]$ | %Out-of-Range Testing |
|---|---|---|---|---|
| **Model A** | 563.6<br>(719.5) | 8.3<br>(11.5) | 693.5 | 12.5 |
| **Model C** | 394.6<br>(565) | 5<br>(8.3) | 516.6 | 8.3 |
| **Model D** | 494.8<br>(658.8) | 7.4<br>(12.4) | 595.3 | 10.1 |
| **Model E** | 1293.7<br>(1414.3) | 35.2<br>(36.3) | 1357.3 | 38.2 |

The metrics on the test set in Model C and Model D, and their corresponding parity plots in Figure 11 show that Model C outperforms Model D, which is expected as the hybrid series configuration allows the model to map the mechanistic model parameters to the input variables efficiently. Hence, these parameters will become dependent upon those variables, maintaining the main functional form in the mechanistic mode. While in the hybrid parallel configuration, those parameters are fixed for the entire dataset, and an error term corrects the prediction of the mechanistic model. The hybrid model configurations (Models C and D) demonstrate slightly more accurate predictions on the test set compared to Model A. However, one might argue that the difference is less than 180 $W/m^2K$ (2.5% of the average HTC, 7,094.8 $W/m^2K$) in the test set, and such a small improvement cannot be justified considering the implementation challenges of a hybrid model. Hence, Model A (pure ML) might be sufficient for predicting the HTC if a systematic data split is not to be used.

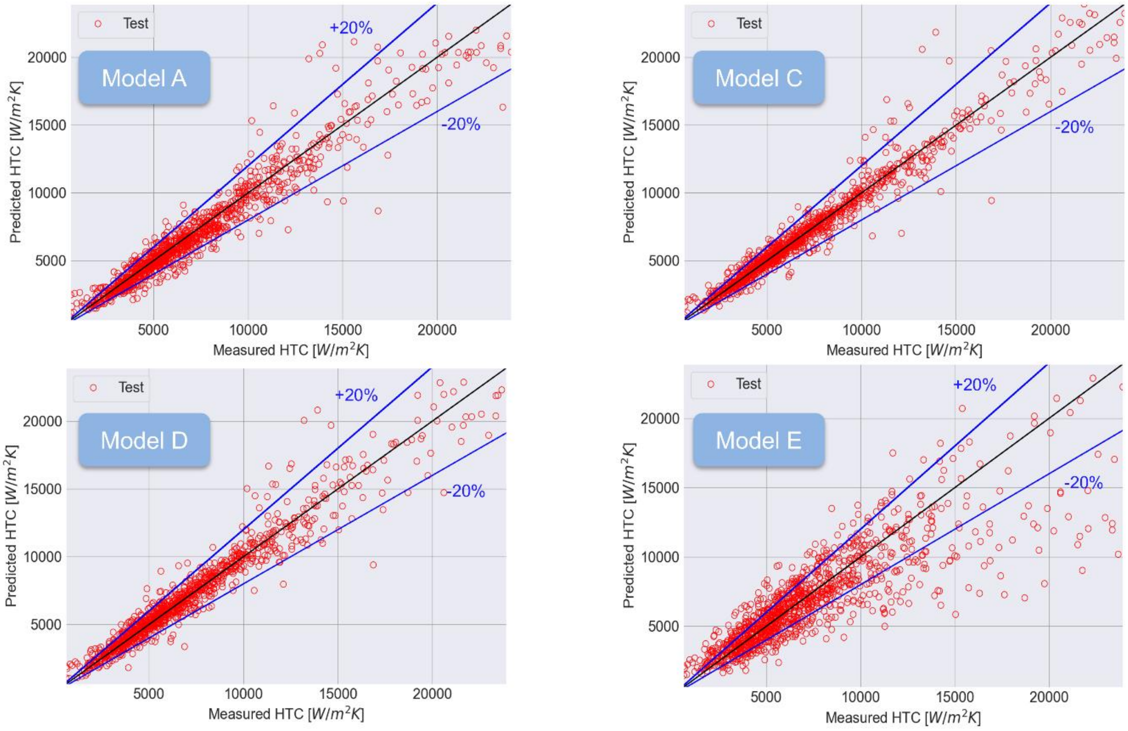


**Figure 11:** Parity plots of the HTC for A) Pure ML, C) Hybrid-series, D) Hybrid-parallel, and E) Pure mechanistic models in Scenario I.

Moreover, the performance of the models is assessed with a sensitivity analysis similar to the one carried out by Loyola-Fuentes et al. (2024) [21] varying the vapour quality ($x$) for various mass flux ($G$) values of R134a. As illustrated in Figure 12, all models show that with increasing vapour quality and mass flux, the predicted HTC increases, a behaviour that is expected from the physics of the system and the experimental data.

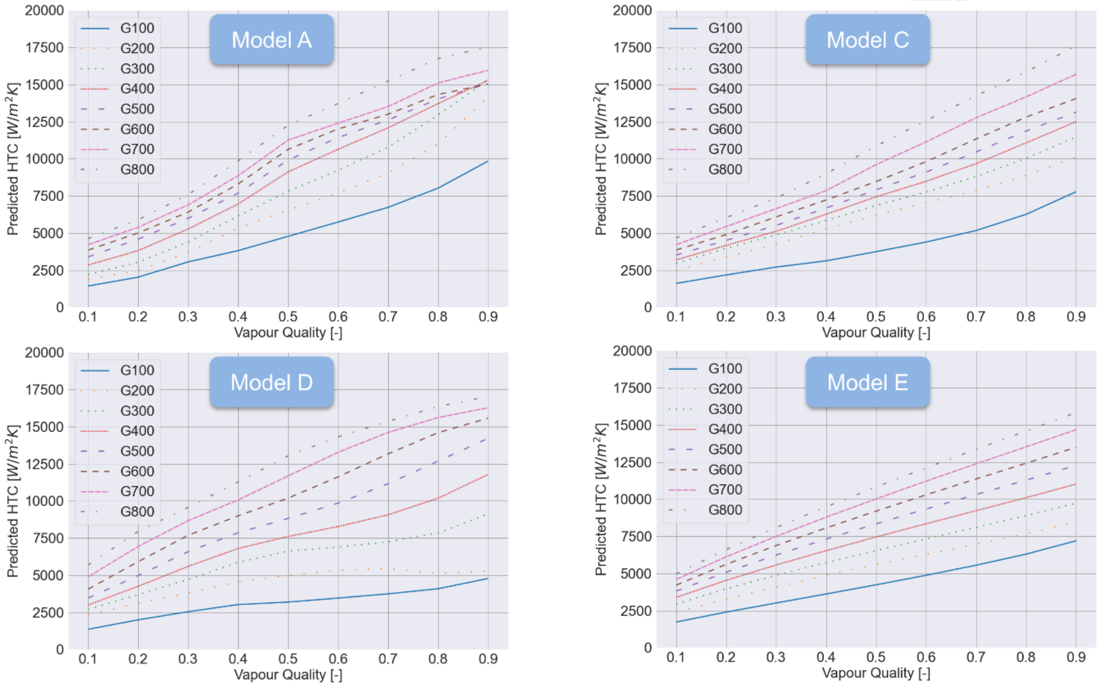


**Figure 12:** Estimation of the HTC for a single tube using A) Pure ML, C) Hybrid-series, D) Hybrid-parallel, and E) Pure mechanistic models in Scenario I.

Now that the sensitivity of all the developed models in this scenario is aligned with the system's expected behaviour, it is essential to understand which model resembles the actual behaviour more closely. To this aim, the heat transfer enhancement factor (EF) is analysed for the experimental data collected for R134a. The EF is defined as the ratio between the measured HTC for the microfin tube and the equivalent HTC of the smooth tube predicted by the mechanistic model. The mechanistic model considers the geometry enhancement factor ($R_x$), and it sets a transition between the gravity-controlled and the forced convective-controlled condensation with a criterion defined on the basis of the refrigerant two-phase flow pattern observations. Thus, the EF can also reveal the significance of the transition between flow patterns across different mass fluxes.  The heat transfer enhancement factor for R134a in the microfin tubes is illustrated in Figure 13 using the collected experimental data, showing that this parameter is relatively constant for $G = 400\ \mathrm{kg/m^2s}$ and $G = 800\ \mathrm{kg/m^2s}$. Thus, considering that the geometry enhancement factor is approximately 1.63, it can be stated that for $G = 400\ \mathrm{kg/m^2s}$, the heat transfer EF is merely due to the geometry of microfin tubes, while the EF lies below the geometrical enhancement factor for $G = 800\ \mathrm{kg/m^2s}$, which can be an indication of a sub-optimal geometry that favours more pressure drop than enhancement in heat transfer. However, the EF shows a completely different behaviour for $G = 100\ \mathrm{kg/m^2s}$ and $G = 200\ \mathrm{kg/m^2s}$ compared to higher mass fluxes. A plausible explanation for this particular behaviour can be the two-phase flow patterns as discussed by Cavallini et al. (2006) [73] and Doretti et al. (2013) [74]. In fact, the microfin tubes promote an earlier transition from stratified flow (i.e. gravity-dominated heat transfer) to the more effective annular flow (i.e. forced convective condensation). As described by Cavallini et al. (2006) [73] and Doretti et al. (2013) [74], this transition occurs at around 200 kg /m$^2$ s and, at these operating conditions, the microfin tubes present the maximum heat transfer enhancement. This is confirmed by the EF profiles reported in Figure 13, in which it can be clearly seen that at $G = 200\ \mathrm{kg/m^2s}$ there is a steep increase in the EF for $x \geq 0.5$, with a maximum value of $EF \approx 2.6$. This should translate into a steeper increase of the HTC for $x \geq 0.6$, while increasing the mass flux from 100 to 200 $\mathrm{kg/m^2s}$, compared to other mass flux values.

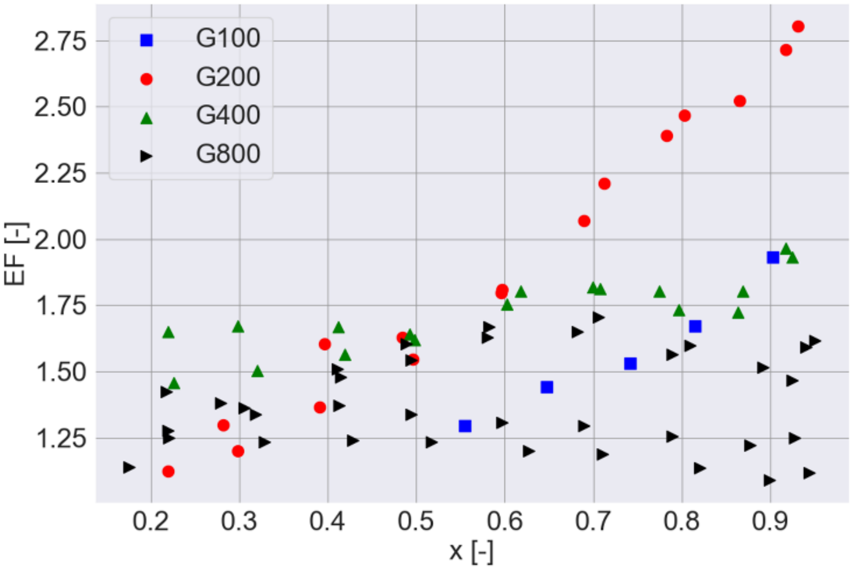


**Figure 13:** Heat transfer enhancement factor versus vapour quality for 100, 200, 400, and 800 $\mathrm{kg/m^2s}$ mass flux with R134a as the refrigerant.

As can be seen in Figure 12, Model E (Mechanistic model) shows that the HTC increases with a relatively similar rate, while increasing $G$. Model D illustrates that for $G = 200\ \mathrm{kg/m^2s}$, the heat transfer coefficient reaches plateau when $x \geq 0.6$, close to the values predicted for $G = 100\ \mathrm{kg/m^2s}$. This is not expected due to the much higher EF observed for $G = 200\ \mathrm{kg/m^2s}$. On the other hand, in this configuration, the HTC sensitivity result for $G \geq 300\ \mathrm{kg/m^2s}$ is as expected, considering that a smoother linear behaviour is observed. Although Model A captures the effect of flow pattern transition for low mass flux, the HTC for larger mass fluxes do not show a consistent behaviour. Among the model candidates presented in Figure 12 only Model C has properly resembled the expected behaviour of the phenomenon, including the impact of the two-phase flow pattern transition. It can be stated that machine learning improved the capability of the mechanistic model to adhere to the non-linear behaviour of the two-phase heat transfer phenomenon.

- **Scenario II**

In this scenario, comparing the performance metrics of the test set with training and validation indicates that, as expected, the mechanistic model (Model E) extrapolates quite. However, this is not the case for Models A, C, and D, which have an ML component. In these models, the error on the training and validation is much lower than the error on the test, especially for Model A. This is an indication of overfitting the training datapoint, as it only enables the model to learn the variations in the data for $G \leq 600\ \mathrm{kg/m^2}s$ during the training loop. The parity plot of Model A in Figure 14 also shows that the model tends to systematically underestimate $HTC > 8{,}000\ \mathrm{W/m^2}$K, leaving almost 43% of the data outside the range of $\pm$20% error in total. Moreover, by using a pure ML model, the error increases by approximately 55% and the out-of-range data by almost 23%. Hence, the mechanistic model is superior to the pure ML model. The main advantage of the hybrid configurations lies in this scenario, as they clearly improve the extrapolation capabilities of the pure ML, while maintaining the prediction accuracy. Among the models developed in this scenario, Model C remains the best candidate as it offers the lowest testing error of 1,092.1 $\mathrm{W/m^2}$K and 12.9% out-of-range data. However, one must note that the difference between Model D and Model C is not significant at all. Their parity plots in Figure 14 and their metrics are shown in Table 12 are evidence to this observation. In the following, the sensitivity analysis assists in selecting the best model candidate more effectively.

**Table 12:** Model performance results for Scenario II, including MAE and out-of-range data

| Model | MAE Training (Validation) [$\mathrm{W/m^2}$K] | %Out-of-Range Training (Validation) | MAE Testing [$\mathrm{W/m^2}$K] | %Out-of-Range Testing |
|---|---|---|---|---|
| **Model A** | 502.4<br>(555.6) | 8.9<br>(10.7) | 2,346 | 42.9 |
| **Model C** | 353.6<br>(466.9) | 4.2<br>(6.8) | 1,092.1 | 12.9 |
| **Model D** | 454.6<br>(516.9) | 7.4<br>(8.3) | 1,269.8 | 14.5 |
| **Model E** | 1296.5<br>(1288.3) | 37.9<br>(37.5) | 1,508.7 | 20.1 |

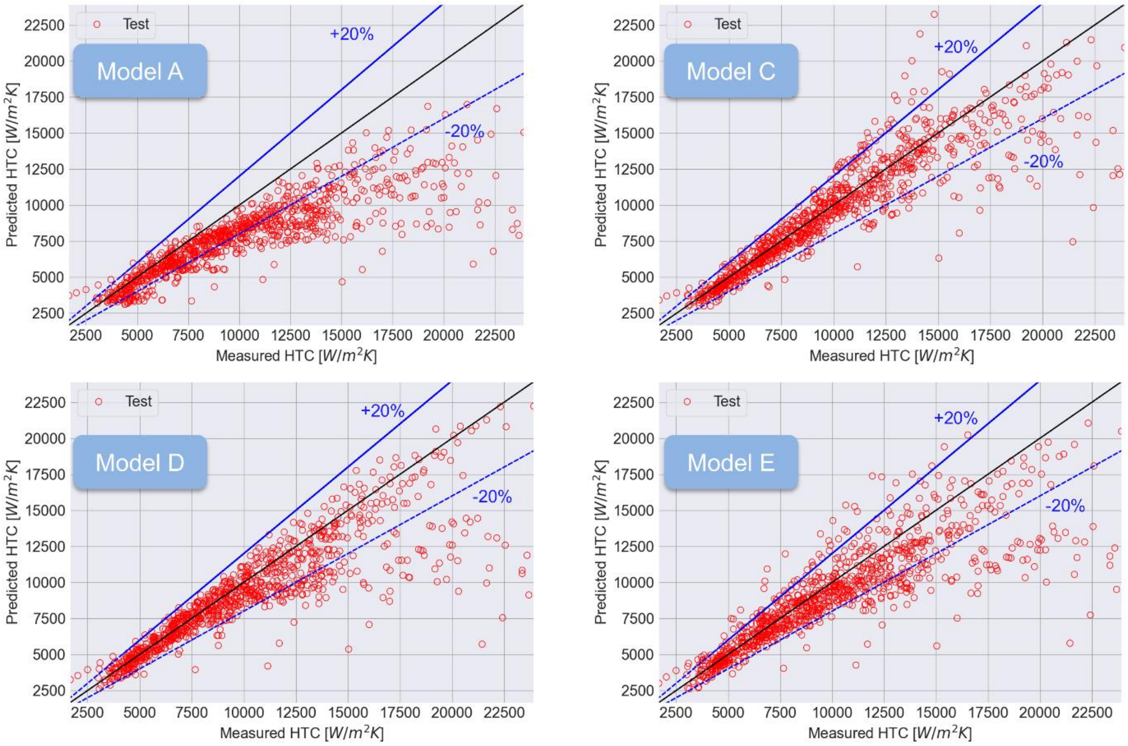


**Figure 14:** Parity plots of the HTC for A) Pure ML, C) Hybrid-series, D) Hybrid-parallel, and E) Pure mechanistic models in Scenario 2.

Figure 15 represents the sensitivity analysis results for Scenario II. Although a different set of data is used for training and validation in this scenario, Model E shows similar behaviour to Scenario I, by neglecting the effect of the two-phase flow pattern transition. Moreover, it is clear from the analysis that Model A systematically underestimates the HTC, since the profiles at high mass flow rates almost overlap, exhibiting similar condensation heat transfer coefficients. The sensitivity results of these two models are additional evidence that neither of them is a proper model candidate for this scenario. On the other hand, Models C and D can capture the effect of EF for $G = 200\ \mathrm{kg/m^2s}$, with Model D as a more reasonable candidate. In our opinion, since Model C is more flexible, the effect of EF for low mass fluxes is propagated into larger mass fluxes as well. The effect is due to the selection of training data, which is dominated by low mass fluxes. However, Model D has a harder constraint that prevents this propagation, as represented in Figure 15, as it uses the mechanistic model with its optimised parameters in parallel with the ML component.

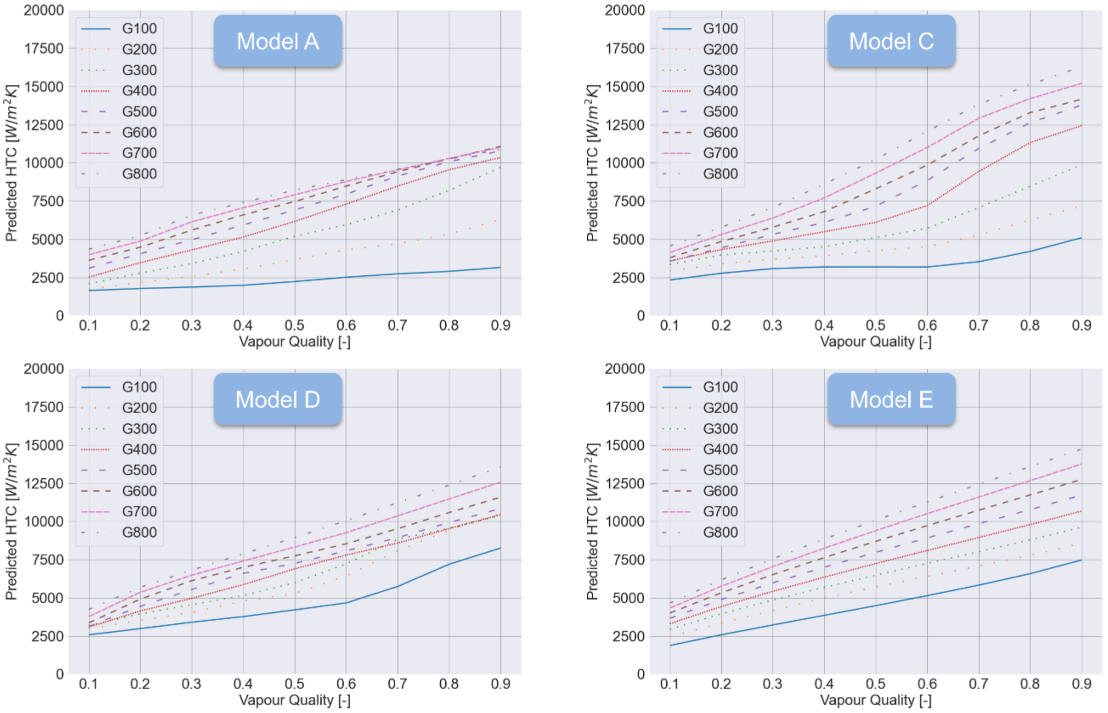


**Figure 15:** Estimation of the HTC for a single tube using A) Pure ML, C) Hybrid-series, D) Hybrid-parallel, and E) Pure mechanistic models in Scenario 2.

## 6 Conclusions

In this study, a systematic hybrid modelling framework has been developed, complementing the developed ML framework for regression of heat transfer data by Loyola-Fuentes et al. (2024) [21], allowing for the selection of various model configurations based on the availability of a mechanistic model and data. The framework has been validated against experimental data of condensation heat transfer coefficient in microfin tubes, by analysing the prediction accuracy and behaviour of the developed models, both for interpolation and extrapolation purposes. It has been demonstrated that in both scenarios, the hybrid models outperformed the mechanistic and pure ML counterparts. However, the proper hybrid configuration depended upon the splitting strategy. In Scenario I, when interpolating, Model C (hybrid series) showed slightly higher accuracy by 2.5% of the average HTC compared to Model D (hybrid parallel). However, the sensitivity analysis showed that Model C captured the physics of the system (i.e. EF effect for $G \leq 200\ \mathrm{kg/m^2s}$) much more effectively than the other candidate. Nevertheless, in Scenario II, this effect was captured more robustly in Model D, although this model had a slightly larger error compared to the hybrid series one. Such behaviour was justified considering that the hybrid series is more flexible compared to the hybrid parallel configuration, as the latter used the optimised mechanistic model in parallel with the ML model. It must be concluded that the hybrid models, in general, outperformed the mechanistic and machine learning models. However, selecting the right configuration is subject to implementation and analysis of the results. As future work, to expand the applicability of the developed modelling framework to dynamic systems such as fouling deposition in heat exchangers, adaptation must be carried out on the proposed framework in this study. As fouling persists as a challenge in heat transfer systems, a hybrid modelling approach can offer a strategic solution by exploiting both the potential of mechanistic and data-driven models.

## References

[1] J. Pinto, M. Mestre, J. Ramos, R.S. Costa, G. Striedner, R. Oliveira, A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks, Comput Chem Eng 165 (2022). https://doi.org/10.1016/j.compchemeng.2022.107952.

[2] R.F. Nielsen, N. Nazemzadeh, L.W. Sillesen, M.P. Andersson, K. V. Gernaey, S.S. Mansouri, Hybrid machine learning assisted modelling framework for particle processes, Comput Chem Eng 140 (2020) 106916. https://doi.org/10.1016/j.compchemeng.2020.106916.

[3] C.A.O. Nascimento, R. Giudici, N. Scherbakoff, Modeling of industrial nylon-6,6 polymerization process in a twin-screw extruder reactor. II. Neural networks and hybrid models, J Appl Polym Sci 72 (1999) 905–912. https://doi.org/10.1002/(SICI)1097-4628(19990516)72:7<905::AID-APP6>3.0.CO;2-7.

[4] H.C. Aguiar, R.M. Filho, Neural network and hybrid model: A discussion about different modeling techniques to predict pulping degree with industrial data, Chem Eng Sci 56 (2001) 565–570. https://doi.org/10.1016/S0009-2509(00)00261-X.

[5] L. Chen, Y. Hontoir, D. Huang, J. Zhang, A.J. Morris, Combining first principles with black-box techniques for reaction systems, Control Eng Pract 12 (2004) 819–826. https://doi.org/10.1016/j.conengprac.2003.09.006.

[6] M. Rudolph, S. Kurz, B. Rakitsch, Hybrid modeling design patterns, J Math Ind 14 (2024). https://doi.org/10.1186/s13362-024-00141-0.

[7] K. Zuo, H.P. Cheng, S.C. Wu, W.T. Wu, A hybrid model combining hydrodynamic and biological effects for production of bacterial cellulose with a pilot scale airlift reactor, Biochem Eng J 29 (2006) 81–90. https://doi.org/10.1016/j.bej.2005.02.020.

[8] Á.J.M. Boareto, M.B. De Souza, F. Valero, B. Valdman, A hybrid neural model (HNM) for the on-line monitoring of lipase production by Candida rugosa, Journal of Chemical Technology & Biotechnology 82 (2007) 319–327. https://doi.org/10.1002/jctb.1678.

[9] N. Nazemzadeh, A.A. Malanca, R.F. Nielsen, K. V. Gernaey, M.P. Andersson, S.S. Mansouri, Integration of first-principle models and machine learning in a modeling framework: An application to flocculation, Chem Eng Sci 245 (2021) 116864. https://doi.org/10.1016/j.ces.2021.116864.

[10] N. Nazemzadeh, An integrated multi-scale modeling framework for flocculation processes, PhD Thesis, Technical University of Denmark, 2022. https://orbit.dtu.dk/en/publications/an-integrated-multi-scale-modeling-framework-for-flocculation-pro (accessed May 1, 2025).

[11] M.L. Thompson, M.A. Kramer, Modeling chemical processes using prior knowledge and neural networks, AIChE Journal 40 (1994) 1328–1340. https://doi.org/10.1002/aic.690400806.

[12] S. Zhang, F. Wang, D. He, R. Jia, Batch-to-batch control of particle size distribution in cobalt oxalate synthesis process based on hybrid model, Powder Technol 224 (2012) 253–259. https://doi.org/10.1016/j.powtec.2012.03.001.

[13] D.Q. Kern, Process heat transfer, McGraw-Hill Inc., 1950.

[14] G. Hewitt, ed., Heat Exchanger Design Handbook 2008, Begell House, 2008.

[15] F.W. Dittus, L.M.K. Boelter, Heat transfer in automobile radiators of the tubular type, International Communications in Heat and Mass Transfer 12 (1985) 3–22. https://doi.org/10.1016/0735-1933(85)90003-X.

[16] J. Ma, Y.P. Huang, J. Huang, Y.L. Wang, Q.W. Wang, Experimental investigations on single-phase heat transfer enhancement with longitudinal vortices in narrow rectangular channel, Nuclear Engineering and Design 240 (2010) 92–102. https://doi.org/10.1016/j.nucengdes.2009.10.015.

[17] S. Aravinth, Prediction of heat and mass transfer for fully developed turbulent fluid flow through tubes, Int J Heat Mass Transf 43 (2000) 1399–1408. https://doi.org/10.1016/S0017-9310(99)00218-5.

[18] G. Righetti, C. Zilio, S. Mancin, G.A. Longo, A review on in-tube two-phase heat transfer of hydro-fluoro-olefines refrigerants, Sci Technol Built Environ 22 (2016) 1191–1225. https://doi.org/10.1080/23744731.2016.1229528.

[19] S.-M. Kim, I. Mudawar, Review of databases and predictive methods for heat transfer in condensing and boiling mini/micro-channel flows, Int J Heat Mass Transf 77 (2014) 627–652. https://doi.org/10.1016/j.ijheatmasstransfer.2014.05.036.

[20] A. Cavallini, D. Del Col, S. Mancin, L. Rossetto, Condensation of pure and near-azeotropic refrigerants in microfin tubes: A new computational procedure, International Journal of Refrigeration 32 (2009) 162–174. https://doi.org/10.1016/j.ijrefrig.2008.08.004.

[21] J. Loyola-Fuentes, N. Nazemzadeh, E. Diaz-Bejarano, S. Mancin, F. Coletti, A framework for data regression of heat transfer data using machine learning, Appl Therm Eng 248 (2024) 123043. https://doi.org/10.1016/j.applthermaleng.2024.123043.

[22] A. Khosravi, J.J.G. Pabon, R.N.N. Koury, L. Machado, Using machine learning algorithms to predict the pressure drop during evaporation of R407C, Appl Therm Eng 133 (2018) 361–370. https://doi.org/10.1016/j.applthermaleng.2018.01.084.

[23] G.M. Hobold, A.K. da Silva, Visualization-based nucleate boiling heat flux quantification using machine learning, Int J Heat Mass Transf 134 (2019) 511–520. https://doi.org/10.1016/j.ijheatmasstransfer.2018.12.170.

[24] B. Kwon, F. Ejaz, L.K. Hwang, Machine learning for heat transfer correlations, International Communications in Heat and Mass Transfer 116 (2020) 104694. https://doi.org/10.1016/j.icheatmasstransfer.2020.104694.

[25] J.-Z. Peng, X. Liu, N. Aubry, Z. Chen, W.-T. Wu, Data-Driven Modeling of Geometry-Adaptive Steady Heat Transfer based on Convolutional Neural Networks: Heat Conduction, ArXiv Preprint (2020).

[26] B. Souayeh, S. Bhattacharyya, N. Hdhiri, M. Waqas Alam, Heat and Fluid Flow Analysis and ANN-Based Prediction of A Novel Spring Corrugated Tape, Sustainability 13 (2021) 3023. https://doi.org/10.3390/su13063023.

[27] J. Loyola-Fuentes, L. Pietrasanta, M. Marengo, F. Coletti, Machine Learning Algorithms for Flow Pattern Classification in Pulsating Heat Pipes, Energies (Basel) 15 (2022) 1970. https://doi.org/10.3390/en15061970.

[28] C. Jayaweera, N. Groot, S. Meul, A. Verliefde, I. Nopens, I. Hitsov, Development of a hybrid model for reliably predicting the thermal performance of direct contact countercurrent cooling towers, Int J Heat Mass Transf 197 (2022) 123336. https://doi.org/10.1016/j.ijheatmasstransfer.2022.123336.

[29] H. Lee, C. Lee, H. Lee, Physics-informed machine learning for enhanced prediction of condensation heat transfer, Energy and AI 20 (2025) 100482. https://doi.org/10.1016/j.egyai.2025.100482.

[30] A. Furlong, X. Zhao, R.K. Salko, X. Wu, Physics-based hybrid machine learning for critical heat flux prediction with uncertainty quantification, Appl Therm Eng 279 (2025) 127447. https://doi.org/10.1016/j.applthermaleng.2025.127447.

[31] C.G. BROYDEN, The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations, IMA J Appl Math 6 (1970) 76–90. https://doi.org/10.1093/imamat/6.1.76.

[32] R. Fletcher, A new approach to variable metric algorithms, Comput J 13 (1970) 317–322. https://doi.org/10.1093/comjnl/13.3.317.

[33] D. Goldfarb, A family of variable-metric methods derived by variational means, Math Comput 24 (1970) 23–26. https://doi.org/10.1090/S0025-5718-1970-0258249-6.

[34] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math Comput 24 (1970) 647–656. https://doi.org/10.1090/S0025-5718-1970-0274029-X.

[35] H. Qi, X.-G. Zhou, L.-H. Liu, W.-K. Yuan, A hybrid neural network-first principles model for fixed-bed reactor, Chem Eng Sci 54 (1999) 2521–2526. https://doi.org/10.1016/S0009-2509(98)00523-5.

[36] H. Yu, B.M. Wilamowski, Levenberg–Marquardt Training, in: Intelligent Systems, CRC Press, 2018: pp. 12-1-12–16. https://doi.org/10.1201/9781315218427-12.

[37] P. Lauret, H. Boyer, J.C. Gatina, Hybrid Modelling of the Sucrose Crystal Growth Rate, International Journal of Modelling and Simulation 21 (2001) 23–29. https://doi.org/10.1080/02286203.2001.11442183.

[38] P. Jul-Rasmussen, M. Kumar, J. Pinto, R. Oliveira, X. Liang, J.K. Huusom, Incorporating first-principles information into hybrid modeling structures: Comparing hybrid semi-parametric models with Physics-Informed Recurrent Neural Networks, Comput Chem Eng 199 (2025) 109119. https://doi.org/10.1016/j.compchemeng.2025.109119.

[39] F. Francis-Xavier, F. Kubannek, R. Schenkendorf, Hybrid process models in electrochemical syntheses under deep uncertainty, Processes 9 (2021) 1–17. https://doi.org/10.3390/pr9040704.

[40] D. Levišauskas, V. Galvanauskas, G. Žunda, &s Grigiškis, Model-based optimization of biosurfactant production in fed-batch culture Azotobacter vinelandii, 2004.

[41] Y. Meng, S. Yu, J. Zhang, J. Qin, Z. Dong, G. Lu, H. Pang, Hybrid modeling based on mechanistic and data-driven approaches for cane sugar crystallization, J Food Eng 257 (2019) 44–55. https://doi.org/10.1016/j.jfoodeng.2019.03.026.

[42] R.F. Nielsen, Hybrid Modelling Strategies for Development of Digital Twins, PhD Thesis, Technical University of Denmark, 2021. https://orbit.dtu.dk/en/publications/hybrid-modelling-strategies-for-development-of-digital-twins (accessed July 3, 2025).

[43] M. Alsubeihi, A. Jessop, B. Moseley, C.P. Fonte, A.K. Rajagopalan, Modern, Efficient, and Differentiable Transport Equation Models Using JAX: Applications to Population Balance Equations, Ind Eng Chem Res 64 (2025) 4541–4553. https://doi.org/10.1021/acs.iecr.4c04208.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, Adv Neural Inf Process Syst 32 (2019).

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, (2015). www.tensorflow.org.

[46] DeepMind. The DeepMind JAX Ecosystem, Https://Github.Com/Google-Deepmind/Optax (2020).

[47] A.R.N. Aouichaoui, S.S. Mansouri, J. Abildskov, G. Sin, Uncertainty estimation in deep learning-based property models: Graph neural networks applied to the critical properties, AIChE Journal 68 (2022) 1–15. https://doi.org/10.1002/aic.17696.

[48] V.P. Ramesh, P. Baskaran, A. Krishnamoorthy, D. Damodaran, P. Sadasivam, Back propagation neural network based big data analytics for a stock market challenge, Commun Stat Theory Methods 48 (2019) 3622–3642. https://doi.org/10.1080/03610926.2018.1478103.

[49] A. Cavallini, D. Censi, D. Col, L. Doretti, L. Rossetto, G.A. Longo, Heat transfer coefficients of HFC refrigerants during condensation at high temperature inside an enhanced tube, in: International Refrigeration and Air Conditioning Conference, Purdue, 2002.

[50] A. Cavallini, G. Censi, D. Del Col, L. Doretti, G.A. Longo, L. Rossetto, C. Zilio, Experimental heat transfer coefficient and pressure drop during condensation of R22 and R407C inside a horizontal microfin tube, in: Proceeding of International Heat Transfer Conference 12, Begellhouse, Connecticut, 2002. https://doi.org/10.1615/IHTC12.1190.

[51] A. Cavallini, In-tube condensation performance of refrigerants, in: 11th Proc. of Int. Refrigeration and Air Conditioning Conf. at Purdue, Purdue, 2006.

[52] H. Haraguchi, Condensation of pure refrigerants in horizontal smooth and microfin tubes, Dr. Eng., Kyushu University, 1994.

[53] M.A. Kedzierski, J.M. Goncalves, Horizontal Convective Condensation of Alternative within a Micro-Fin Tube, 1997.

[54] M.-H. Kim, J.-S. Shin, Condensation heat transfer of R22 and R410A in horizontal smooth and microfin tubes, International Journal of Refrigeration 28 (2005) 949–957. https://doi.org/10.1016/j.ijrefrig.2005.01.017.

[55] J. Zilly, J. Jang, P.S. Hrnjak, Condensation of CO 2 at Low Temperature inside Horizontal Microfinned Tubes, 2003.

[56] L. Colombo, A. Lucchini, A. Muzzio, A. Niro, HEAT TRANSFER AND PRESSURE DROP DURING EVAPORATION AND CONDENSATION OF R134a INSIDE TWO MICROFIN TUBES OF NEW DESIGN, in: Multiphase, Begell House Inc., 2006. https://doi.org/10.1615/IHTC13.p12.100.

[57] M. Uchida, M. Itoh, N. Shikazono, T. Hatada, T. Ohtani, Development of high performance microfin tubes (2nd report: experimental study), in: 31st Cond. and Refrig. Eng. Joint Conference, Tokyo, 1997: pp. 81–84.

[58] A. Cavallini, D. Del Col, L. Doretti, M. Matkovic, L. Rossetto, C. Zilio, G. Censi, Condensation in Horizontal Smooth Tubes: A New Heat Transfer Model for Heat Exchanger Design, Heat Transfer Engineering 27 (2006) 31–38. https://doi.org/10.1080/01457630600793970.

[59] J. Zhang, W. Li, W.J. Minkowycz, Numerical simulation of R410A condensation in horizontal microfin tubes, Numeri Heat Transf A Appl 71 (2017) 361–376. https://doi.org/10.1080/10407782.2016.1243934.

[60] A. Diani, A. Cavallini, L. Rossetto, R1234yf condensation inside a 3.4 mm ID horizontal microfin tube, International Journal of Refrigeration 75 (2017) 178–189. https://doi.org/10.1016/j.ijrefrig.2016.12.014.

[61] A. Diani, P. Brunello, L. Rossetto, R513A condensation heat transfer inside tubes: Microfin tube vs. smooth tube, Int J Heat Mass Transf 152 (2020) 119472. https://doi.org/10.1016/j.ijheatmasstransfer.2020.119472.

[62] A. Diani, M. Campanale, A. Cavallini, L. Rossetto, Low GWP refrigerants condensation inside a 2.4 mm ID microfin tube, International Journal of Refrigeration 86 (2018) 312–321. https://doi.org/10.1016/j.ijrefrig.2017.11.011.

[63] G.A. Longo, S. Mancin, G. Righetti, C. Zilio, Comparative analysis of microfin vs smooth tubes in R32 and R410A condensation, International Journal of Refrigeration 128 (2021) 218–231. https://doi.org/10.1016/j.ijrefrig.2021.04.011.

[64] G. Li, L. Huang, L. Tao, Experimental investigation of refrigerant condensation heat transfer characteristics in the horizontal microfin tubes, Appl Therm Eng 123 (2017) 1484–1493. https://doi.org/10.1016/j.applthermaleng.2017.05.080.

[65] M.K. Bashar, K. Nakamura, K. Kariya, A. Miyara, Condensation heat transfer of R1234yf in a small diameter smooth and microfin tube and development of correlation, International Journal of Refrigeration 120 (2020) 331–339. https://doi.org/10.1016/j.ijrefrig.2020.09.002.

[66] A. Diani, M. Campanale, L. Rossetto, Experimental study on heat transfer condensation of R1234ze(E) and R134a inside a 4.0 mm OD horizontal microfin tube, Int J Heat Mass Transf 126 (2018) 1316–1325. https://doi.org/10.1016/j.ijheatmasstransfer.2018.06.047.

[67] C. Kondou, Heat transfer and pressure drop of R1123/R32 (40/60 mass%) flow in horizontal microfin tubes during condensation and evaporation, Sci Technol Built Environ 25 (2019) 1281–1291. https://doi.org/10.1080/23744731.2019.1642096.

[68] L. Qingpu, C. Guangming, W. Qin, T. Leren, X. Yongmei, Experimental study of condensation heat transfer of R134a inside the micro-fin tubes at high mass flux, Int J Heat Mass Transf 187 (2022) 122524. https://doi.org/10.1016/j.ijheatmasstransfer.2022.122524.

[69] L. Ma, X. Liu, Y. Gao, W. Li, Z. Wu, X. Luo, Z. Tao, S. Kabelac, R410A and R32 condensation heat transfer and flow patterns inside horizontal micro-fin and 3-D enhanced tubes, International Communications in Heat and Mass Transfer 142 (2023) 106638. https://doi.org/10.1016/j.icheatmasstransfer.2023.106638.

[70] M. Hirose, D. Jige, N. Inoue, Optimum fin geometries on condensation heat transfer and pressure drop of R1234ze(E) in 4-mm outside diameter horizontal microfin tubes, Sci Technol Built Environ 25 (2019) 1271–1280. https://doi.org/10.1080/23744731.2019.1622960.

[71] M. Hirose, J. Ichinose, N. Inoue, Development of the general correlation for condensation heat transfer and pressure drop inside horizontal 4 mm small-diameter smooth and microfin tubes, International Journal of Refrigeration 90 (2018) 238–248. https://doi.org/10.1016/j.ijrefrig.2018.04.014.

[72] C.-H. Kim, N.-H. Kim, Condensation heat transfer and pressure drop of low GWP R-404A-alternative refrigerants (R-448A, R-449A, R-455A, R-454C) in a 7.0-mm outer-diameter horizontal microfin tube, International Journal of Refrigeration 126 (2021) 181–194. https://doi.org/10.1016/j.ijrefrig.2021.01.025.

[73] A. Cavallini, D. Del Col, L. Doretti, S. Mancin, L. Rossetto, C. Zilio, Visualization of the Heat Transfer Enhancement During Condensation in a Microfin Tube, in: Volume 2: Fora, ASMEDC, 2006: pp. 685–690. https://doi.org/10.1115/FEDSM2006-98514.

[74] L. Doretti, C. Zilio, S. Mancin, A. Cavallini, Condensation flow patterns inside plain and microfin tubes: A review, International Journal of Refrigeration 36 (2013) 567–587. https://doi.org/10.1016/j.ijrefrig.2012.10.021.

## Acknowledgements

The authors acknowledge Hexxcell Ltd. for providing the funding for the work presented here.

## Nomenclature

***Abbreviations***

| | |
|---|---|
| *AD* | Automatic differentiation |
| *ANN* | Artificial neural network |
| *Adam* | Adaptive moment estimation method |
| *BFGS* | Broyden-Fletcher-Goldfarb-Shanno |
| *CV* | Cross-validation |
| *EF* | Enhancement factor |
| *GWP* | Global warming potential |
| *HFC* | Hydrofluorocarbon |
| *HTC* | Heat transfer coefficient |
| *LM* | Levenberg-Marquardt |
| *MAE* | Mean absolute error |
| *ML* | Machine learning |
| *MSE* | Mean squared error |
| *ND* | Numerical differentiation |
| *PBE* | Population balance equation |
| *PLS* | Partial least squares |
| *PSD* | Particle size distribution |
| *RBFN* | Radial basis function network |
| *RF* | Random forest |
| *RFECV* | Recursive feature elimination with cross-validation |
| *SD* | Symbolic differentiation |
| *SVM* | Support vector machine |

***Symbols***

| | |
|---|---|
| *Bo* | Bond number $[-]$ |
| *cp* | Heat capacity [J $\mathrm{kg}^{-1}$ $\mathrm{K}^{-1}$] |
| $\boldsymbol{\mathcal{D}}$ | Experimental data |
| $\bar{\boldsymbol{\mathcal{D}}}$ | Pre-processed data |
| $d_f$ | Fin tip diameter [m] |
| *f(.)* | Mechanistic model |
| $J_V$ | Vapour velocity $[-]$ |
| $\mathcal{L}$*(.)* | Machine learning loss function |
| *G* | Refrigerant mass flux [kg $\mathrm{m}^{-2}\mathrm{s}^{-1}$] |
| *h(.)* | Machine learning model |
| $h_f$ | Fin height $[m]$ |
| $h_{LV}$ | Heat of vaporisation [J $\mathrm{kg}^{-1}$] |
| $L_f$ | Tube length [m] |
| $n_f$ | Number of fins per tube $[-]$ |
| $\boldsymbol{\mathscr{p}}(.)$ | Vector of pre-processing functions |
| $P_{red}$ | Reduced pressure $[-]$ |
| $Pr$ | Prandtl number $[-]$ |
| $R_x$ | Geometrical enhancement factor $[-]$ |
| $T$ | Temperature [K] |
| $\Delta T_{dimless}$ | Dimensionless temperature $[-]$ |
| $\boldsymbol{X}$ | Independent variables |
| $x$ | Vapour quality $[-]$ |
| $\boldsymbol{y}$ | Target variables |
| $\hat{\boldsymbol{y}}$ | Estimated target variables |

***Greeks***

| | |
|---|---|
| $\beta$ | Helix angle [°] |
| $\gamma$ | Apex angle [°] |
| $\boldsymbol{\vartheta}$ | ML model parameters |
| $\lambda$ | Thermal conductivity [W $\mathrm{m}^{-1}\mathrm{K}^{-1}$] |
| $\mu$ | Viscosity [Pa s] |
| $\rho$ | Density [kg $\mathrm{m}^{-3}$] |
| $\sigma$ | Liquid surface tension [N $\mathrm{m}^{-1}$] |
| $\tau_c$ | Critical threshold $[-]$ |

***Subscripts***

| | |
|---|---|
| *L* | Liquid |
| *S* | Surface |
| *V* | Vapour |
| *W* | Wall |

***Superscripts***

| | |
|---|---|
| *opt* | optimised |
| *test* | Testing data set |
| *train* | Training data set |
| *true* | True value measured |
| *val* | Validation dataset |

**Declaration of Interest Statement**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The author is an Editorial Board Member/Editor-in-Chief/Associate Editor/Guest Editor for this journal and was not involved in the editorial review or the decision to publish this article.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

None