Can continuous-time ML models outperform discrete-time ML models for heat exchanger dynamic modeling?

Dessie Tadele Embiale[a], Mohamed Tahar Mabrouk[a], Stéphane Grieu[b], Bruno Lacarrière[a]

[a] IMT Atlantique, Department of Energy Systems and Environment, GEPEA, UMR CNRS 6144, Nantes, France, dessie-tadele.embiale@imt-atlantique.fr CA, mohamed-tahar.mabrouk@imt-atlantique.fr, bruno.lacarriere@imt-atlantique.fr

[b] PROMES-CNRS (UPR 8521), Université de Perpignan Via Domitia, rambla de la thermodynamique, Tecnosud, 66100 Perpignan, France, grieu@univ-perp.fr

Abstract:
This study investigates the capability of continuous and discrete-time machine learning models to approximate the dynamic behavior of heat exchangers in a gas-turbine-based Combined Heat and Power (CHP) system. Two continuous-time models are proposed based on the Neural ordinary differential equations (Neural-ODE) framework, one purely data driven Neural-ODE and the second a Neural-ODE with physics embedded in it. The proposed models are compared with a discrete-time Long Short-Term Memory (LSTM) model in terms of training data requirement, prediction accuracy, and computational efficiency. The results indicate that continuous-time models can approximate the predictive performance of the discrete-time LSTM model while requiring significantly less training data. In particular, the physics-embedded model demonstrates strong stability and generalization capability across varying operating conditions, achieving error percentage below 0.452% and R^2 score closer to 0.999. Computationally, the LSTM model is faster to train due to its discrete structure, whereas the continuous-time models require longer training times because they rely on numerical integration through an ODE solver. Overall, the findings highlight that continuous-time models, especially when known physical knowledge is embedded, provide a data-efficient and physically consistent alternative to discrete-time models for modeling dynamic energy systems.

Keywords:
Heat exchangers, Machine learning, Continuous-time models, Discrete-time models, Combined Heat and Power (CHP).

1. Introduction

Accurate and computationally efficient dynamic modeling is crucial for real-time optimization, monitoring, and control of modern multi-energy systems. These systems integrate multiple energy technologies such as thermal, electrical, and chemical conversion processes which work in close synergy to improve overall efficiency, flexibility, and reliability. Heat exchangers are key components in many of these technologies such as power plants [1], gas turbines, heat pumps, and air conditioning and refrigeration systems [2,3]. In many of the applications, heat exchangers are often required to operate across a wide range of dynamic conditions, including frequent load changes, start-up and shut-down sequences, and varying inlet conditions [4,5]. Due to their nonlinear and transient behavior, these heat exchangers often dominate the dynamic behavior of the overall system. Inaccurate or overly simplified dynamic representation can therefore lead to reduced performance and efficiency, necessitating the development of accurate yet computationally efficient dynamic models.

Various modeling approaches have been used to model the dynamics of heat exchangers which can be classified into two classes: physics-based and data-driven methods. Depending on the application and the level of detail required, physics-based models range from simplified lumped parameter

formulations [6] to spatially distributed frameworks [7, 8] that account for variations in temperature, heat flux, and fluid properties along the flow direction. Lumped dynamic models assume spatially uniform temperature and fluid properties, which significantly reduce their computational cost and make them well-suited for real-time simulation, control, and optimization applications. However, this assumption limits the ability of the model to capture temperature gradients along the length of the heat exchanger and spatially varying heat transfer coefficients [9, 10]. On the other hand, distributed parameter models discretize the heat exchanger into a finite number of control volumes, and mass and energy conservation equations are applied to each control volume. This allows distributed parameter models to accurately capture spatially varying heat transfer phenomena, temperature and fluid property variations. However, they are computationally intensive due to the need to solve a large number of coupled energy balance equations [7, 11].

Data-driven modeling offers a flexible alternative to physics-based modeling by capturing system dynamics directly from data. These methods capture the relationship between inputs like flow-rates and inlet temperatures and outputs, such as outlet temperatures and heat transfer rates [12, 13]. They are computationally efficient once trained; however, their performance is limited to the range and quality of available data, and they offer less physical consistency compared to physics-based models. Physics-Informed Neural Networks(PINNs) improve consistency by embedding governing physical laws, expressed as residuals of differential equations, directly into the training process of Neural Networks (NN). This guides the NN towards physically meaningful solutions since it fits not only the data but also the fundamental behavior of the system. The most common applications of ML models for heat exchanger modeling include prediction of outlet temperatures [14], heat transfer coefficient [15, 16], pressure drop [16, 17] and fouling factor [18].

Heat exchangers exhibit dynamic behavior as their thermal performance evolves over time in response to variations in operating conditions. This behavior can be approximated with Recurrent Neural Networks (RNNs), which are a class of deep learning architectures specifically designed to model sequential and time dependent problems [19, 20]. Standard RNNs suffer from vanishing and exploding gradient problems, which limit their ability to capture long term dependencies. Long Short-Term Memory (LSTMs) addresses this with gated memory, making them effective for modeling dynamic systems. At each time step, the network maintains and updates a hidden state that captures temporal dependencies and propagates information across the sequence, enabling the model to retain memory of past inputs when producing future outputs. This makes RNNs naturally suited for learning the dynamic behavior of physical systems directly from data. However, RNNs operate in discrete time, which can limit their ability to generalize across variable time steps and makes it difficult to incorporate a known physical structure into the model. Recently, Neural Ordinary Differential Equations (Neural-ODEs) have emerged as a data-driven modeling tool to capture the continuous-time dynamics of nonlinear systems [21–23]. Unlike RNNs, Neural-ODEs formulate the temporal evolution of the system as a differential equation parameterized by a NN as indicated in (1).

$$\frac{dx(t)}{dt} = f_\theta(x(t), t) \tag{1}$$

where $f_\theta(.)$ represents a NN that learns the underlying vector field directly from data. Rather than updating the hidden state through a discrete recurrence relation, the state is evolved forward in time by numerically integrating the NN parameterized ODE using standard ODE solvers. This continuous-time representation enables Neural-ODEs to naturally handle irregularly sampled data and variable simulation horizons without any modification, as the ODE solver adaptively determines the integration steps required to achieve a specific accuracy. The continuous-time formulation of Neural-ODEs is also more consistent with the governing equations of most engineering systems, which are inherently described by differential equations rather than discrete relations. This makes Neural-ODEs flexible to integration with physics-based models, as the NN can be incorporated alongside known physical terms within a single ODE framework.

Neural-ODEs can be used for modeling heat exchanger dynamics to learn the underlying thermal behavior in continuous time, enabling accurate yet computationally efficient representations suitable for real-time simulation, control, and optimization applications. In this study, two continuous-time-based modeling approaches for single-phase heat exchangers are proposed, both of which leverage the Neural-ODE framework. The first approach is a purely data-driven Neural-ODE model that learns the outlet temperature dynamics entirely from data, while the second is a hybrid model that combines a simplified physical model of the heat exchangers with a NN component, allowing the model to exploit known physical structure while retaining flexibility to capture complex dynamics that the simplified equation alone cannot capture. Both approaches are compared against a discrete-time LSTM model, which serves as a baseline representative of state-of-the-art data-driven modeling of time series.

2. Case study

The proposed models are tested on Combined Heat and Power (CHP) generation system based on a gas turbine, as shown in Fig. 1. It is an externally-fired gas turbine (EFGT) system adapted from [24], in which fuel combustion occurs outside the turbine's working fluid loop, with heat transferred indirectly to the working gas through a high temperature heat exchanger. This separation of the combustion chamber from the system enables fuel flexibility without exposing the turbine to corrosive species, particulates, or ash. The CHP system consists of a compressor, a turbine, and three interconnected single-phase heat exchangers: The recuperator and combustion heat exchanger are gas-gas heat exchangers, and the heat recovery system is a gas-liquid heat exchanger. The primary inputs to the CHP system include inlet temperature and mass flow rate to the compressor, fuel flow rate and inlet air flow rate to the combustion chamber, inlet temperature and mass flow rate of water to the heat recovery and Speed of the shaft. The compressor and turbine are modeled as static components, whereas the dynamic models are used for the heat exchangers. The details of the models used are discussed in the next section.

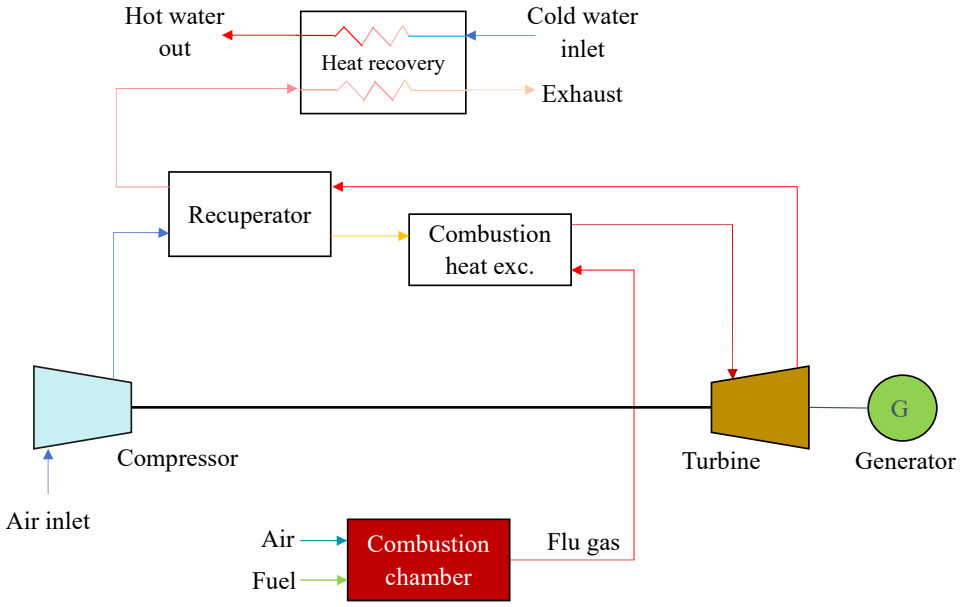

Figure 1: CHP layout.

3. Methodology

The models used for the components of the CHP system and the proposed continuous-time models are presented in this section. After that, the training procedures for both the continuous and discrete-time models are described. This paper focuses on developing surrogate model for the heat exchangers. The other components are kept simple and remained unchanged in the final CHP system, while the heat

exchangers are modeled with a higher detail.

Compressor model

The output parameters of the compressor are outlet temperature and power consumption and are given in (2), and (3), respectively [24].

$$T_{o,co} = T_a \left[1 + \frac{1}{\eta_{co}} \left(r_{co}^{\frac{k-1}{k}} - 1\right)\right] \tag{2}$$

$$\dot{W}_{co} = \dot{m}_{co}\, C_{p,a}\, (T_{o,co} - T_a) \tag{3}$$

Turbine model

The turbine is modeled using inlet temperature and intake mass flow rate as input parameters and outlet exhaust gas temperature and output power as the output parameters [24].

$$T_{o,t} = T_{in,t} \left[1 - \eta_t \left(1 - \left(\frac{1}{r_t}\right)^{\frac{k-1}{k}}\right)\right] \tag{4}$$

$$\dot{W}_t = \dot{m}_{co}\, C_{p,a}\, (T_{t,in} - T_{o,t}) \tag{5}$$

Combustion chamber

The energy balance on the combustion chamber can be given as [25]:

$$\dot{m}_a C_{p_a} T_{in,cc} + \dot{m}_{fuel} LHV = \dot{m}_{fg} C_{p,fg} T_{o,cc} + (1 - \eta_{cc}) \dot{m}_{fuel} LHV \tag{6}$$

3.1. Heat exchanger models

A high-fidelity one dimensional model that takes into account spatial variation of the temperatures and fluid properties along the flow direction is developed and used as a reference for the analysis. Two continuous-time-based models capable of determining the outlet temperatures of the hot and cold fluids based on the Neural-ODE framework are also presented.

One-dimensional (1D) model

Dynamic energy balance equations are used to capture spatial and temporal variations of the temperatures of the hot fluid, cold fluid, and the separating wall. These equations describe space and time dependent evolution of the temperatures by considering convective heat transfer within the hot and cold fluids and the thermal interaction between the three domains. The following assumptions are considered:

- Both fluids are single phase.
- The heat exchanger is assumed to be adiabatic, with negligible heat loss to the surrounding.
- Conduction along the flow axis is negligible and pressure drops in the fluids are not considered.

$$\text{For the hot fluid:} \quad (MC_p)_h \left(\frac{dT_h}{dt} + u_h \frac{dT_h}{dx}\right) = h_h A_h (T_w - T_h) \tag{7}$$

$$\text{For the wall:} \quad (MC_p)_w \frac{dT_w}{dt} = h_h A_h (T_h - T_w) - h_c A_c (T_w - T_c) \tag{8}$$

$$\text{For the cold fluid:} \quad (MC_p)_c \left(\frac{dT_c}{dt} + u_c \frac{dT_c}{dx}\right) = h_c A_c (T_w - T_c) \tag{9}$$

To solve (7), (8) and (9), the heat exchanger is spatially discretized using the finite volume method. After a grid independence test, 80 control volumes are chosen as a compromise between accuracy and efficiency, and the resulting ODEs are solved using DifferentialEquations.jl library's ODE solver.

Neural-ODE

In this approach, a neural network is used to learn the governing outlet temperature dynamics of the heat exchangers. By embedding the NN directly into the differential equation, the model naturally enforces temporal continuity. Let $T_{o,h}$ and $T_{o,c}$ represent the outlet temperatures of the hot and cold fluids, respectively, and $\boldsymbol{u}$ a vector of inputs including inlet temperatures and mass flow rates. The heat exchanger system is formulated as

$$\frac{d\boldsymbol{T}}{dt} = f_\theta(\boldsymbol{T}, \boldsymbol{u}, t) \tag{10}$$

Where $\boldsymbol{T} = [T_{o,h}, T_{o,c}]$ is a vector containing hot and cold output temperatures and f_θ is a learnable function. Typically, neural networks are used for their universal approximation properties.

Physics-Constrained Neural-ODE (PC-Neural-ODE)

In this approach, a known physical dynamics of the system is explicitly embedded into the differential equation, while the NN learns the additional unknown dynamics that the added physics alone cannot capture.

$$\text{For the hot fluid:} \quad \frac{dT_{o,h}}{dt} = \frac{m_{f,h}C_{p,h}(T_{in,h} - T_{o,h}) + f_{\theta,1}(\boldsymbol{T}, \boldsymbol{u}, t)}{MC_p} \tag{11}$$

$$\text{For the cold fluid:} \quad \frac{dT_{o,c}}{dt} = \frac{m_{f,c}C_{p,c}(T_{in,c} - T_{o,c}) + f_{\theta,2}(\boldsymbol{T}, \boldsymbol{u}, t)}{MC_p} \tag{12}$$

3.2. Training of the NN

Training a Neural-ODE involves learning the parameters of a NN that describes the continuous-time dynamics of the system. During training of the pure Neural-ODE, the ODE equation in (10) is numerically solved forward in time, and the resulting predicted outlet temperature trajectories are compared with the corresponding outlet temperature profiles from the 1D model. The discrepancy between the two values is evaluated by measuring their mismatch over a training horizon, and this mismatch is minimized by iteratively updating the NN parameters through gradient-based optimization, with gradients computed via adjoint sensitivity analysis. The NN parameters are first optimized using Adam optimizer and subsequently refined using L-BFGS optimizer, by minimizing the sum of square error between the predicted and actual outlet temperature trajectories. Grid search is used for hyperparameter tuning, and the NN consisting of two hidden layers with 15 neurons each is selected as the optimal architecture. After each layer, hyperbolic tangent function is applied to capture the non-linear dynamics of the system.

The same procedure as for the pure Neural-ODE is used while training the PC-Neural-ODE. In the PC-Neural-ODE, the NN is trained to learn the unknown part of dynamics which is not captured with the added physics. Equations 11 and 12 are solved forward in time and the solution is compared with the corresponding outlet temperature trajectories from the 1D model. The same training procedure as the pure Neural-ODE model is used and after grid search, the size of the NN also remained same.

A discrete-time Long Short-Term Memory (LSTM) recurrent model is trained to predict the hot water outlet temperature, the exhaust gas temperature, and the net mechanical power of the CHP. Unlike the Neural-ODEs, which are continuous-time models, the LSTM is a discrete-time model that learns the mapping between the past and future states. During training, sequences of inputs are fed into the network and the predicted values at each time step are compared with the corresponding values from the 1D model. Primary inputs of the CHP system, such as mass flow rate and inlet water temperature to the heat recovery, fuel flow rate and air temperature to the combustion chamber, and shaft speed and inlet air temperature to the compressor, are fed as inputs to the model. After an extensive hyperparameter tuning process, the LSTM model was trained using three hidden layers with 100 neurons

each, employing tanh as an activation function within each cell. The model parameters were optimized using the Adam optimizer, which adaptively adjusts the learning rates for each parameter to enhance the convergence speed and overall stability of the training.
Training data is synthesized from a high fidelity physics-based CHP model simulation. The model integrates static component models detailed in section 3. for the compressor, turbine and combustion chamber, coupled with 1D model for the heat exchangers detailed in section 3.1.. To ensure sufficient temporal variability and operational coverage, the CHP model was simulated over a time span of 6000 seconds. During each simulation, the primary inputs of the CHP system are randomly varied, as shown in Figs. 2 to 4. This completely random variation exposes the system to different operating conditions and helps to capture a wide range of system behaviors in the data.

4. Results and Discussions

This section presents a comprehensive evaluation of the continuous and discrete-time models developed for the heat exchanger dynamic response prediction. All models were trained and evaluated on identical datasets to ensure a fair and rigorous comparison. The comparison focuses on training data requirement, prediction accuracy, robustness to varying operating conditions including extrapolation out of the training range, and computational efficiency.
The results of both models are first validated against the one-dimensional model by running multiple simulations over a time span of 6000 seconds as used in the data generation. The simulations are then extended to 20000 and 30000 seconds to assess the long-term predictive capability of each model. In all simulations, the primary inputs of the CHP system are randomly varied, as shown in Figs. 2 to 4.
The models are validated with respect to two primary quantities of interest in the CHP system: the

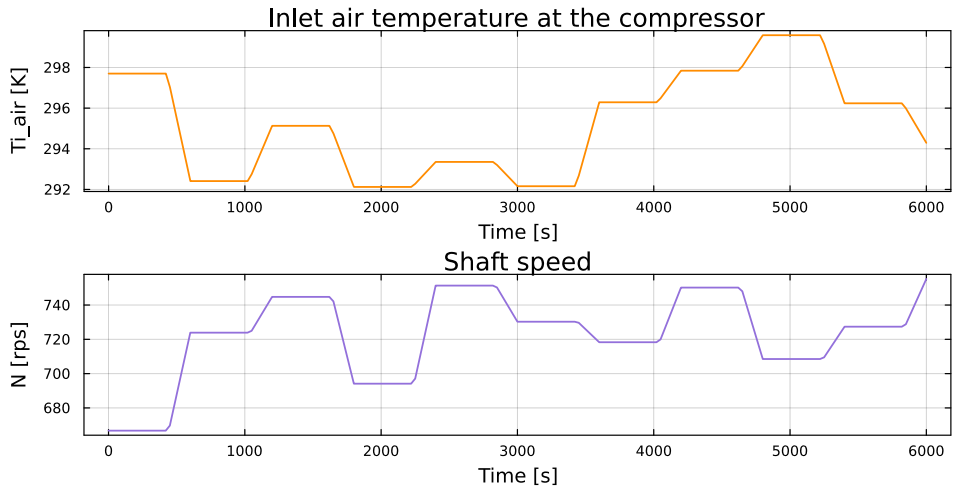

Figure 2: Dynamic variations of shaft speed and inlet air temperature.

net power output and hot water temperature at the outlet of the heat recovery system. The net power output of the CHP is determined as the difference between the turbine output power and the compressor power consumption ($P = \dot{W}_t - \dot{W}_{co}$). Figures 5 and 6 present the results of the first and second validation simulations, respectively. These figures compare the performance of the purely data-driven Neural-ODE, the PC-Neural-ODE, and the LSTM-3 model (LSTM model trained using dataset of 30 simulation sequences of 6000 seconds) against the 1D model for the prediction of power output and hot water temperature. Overall, these results demonstrate the ability of both the continuous-time-based models and the discrete-time-based model to accurately reproduce the dynamic behavior of the system under different operating conditions.

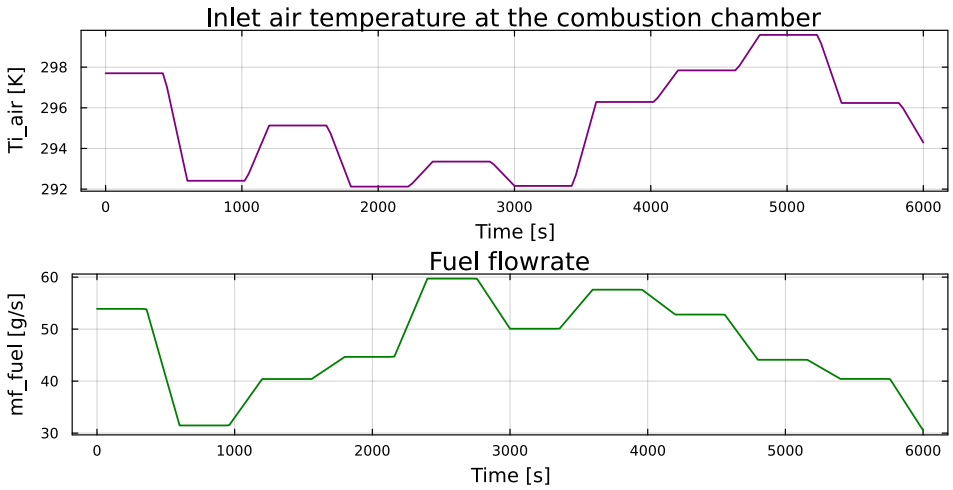

Figure 3: Dynamic variations of primary inputs to the combustion chamber.

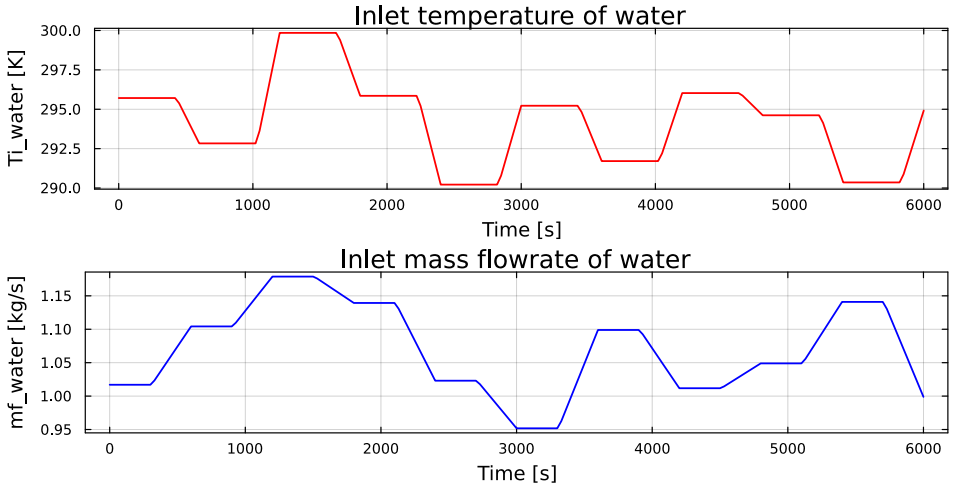

Figure 4: Dynamic variations of inputs to the heat recovery.

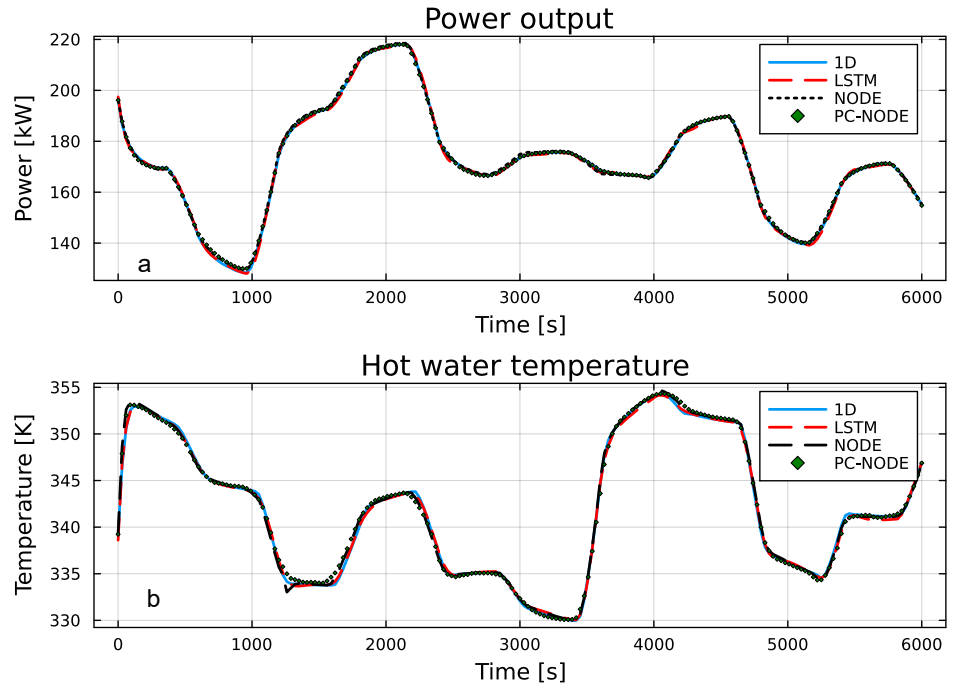

Figure 5: Dynamics of a) power output and b) hot water temperature.

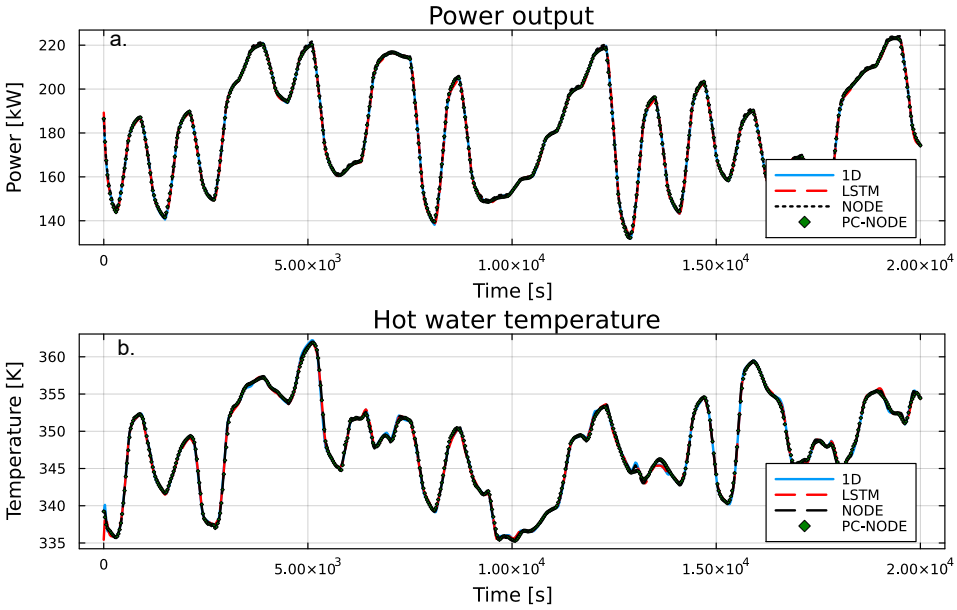

Figure 6: Dynamics of a) power output and b) hot water temperature.

Prediction accuracy

Prediction accuracy is also assessed in terms of quantitative error metrics to provide a comprehensive evaluation of the performance of each model, as defined by (13) and (14).

$$\text{Error } \% = \left(\frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i|} \right) \times 100 \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{14}$$

where y_i and $\hat{y}_i$ denote the true and predicted values, respectively, and n is the number of samples. Table 1 presents a comprehensive comparison of the predictive performances of the purely data-driven Neural-ODE, the PC-Neural-ODE and the LSTM model trained with 12 (LSTM-1), 20 (LSTM-2) and 30 (LSTM-3) simulation datasets against the 1D model. The first validation case lies within the training range in terms of both inputs and simulation time span. It is observed that the continuous-time models achieve errors below 0.31% with $R^2 = 0.999$, indicating that the models accurately capture the dynamics of the power output of the system. Among the discrete-time models, LSTM-3 has the best performance and LSTM-1 shows noticeably larger deviations. When the prediction horizon is extended beyond the training duration (validation simulations 2 (20000s) and 3 (30000s)), only a slight degradation in performance is observed in the purely data driven Neural-ODE and LSTM-3, while the PC-neural-ODE performance remains nearly unchanged. This highlights the benefit of embedding known physical knowledge into the model, enabling robust long-term predictions and improved generalization in scenarios where purely data-driven models might struggle.
The fourth validation simulation evaluates the extrapolation capability with inputs outside of the training data range. The mass flow rate of water is varied between $0.8kg/s$ and $1.3kg/s$, and the fuel flow rate is varied between $20g/s$ and $70g/s$, which are outside the range used during the training phase. In this case, the purely data-driven Neural-ODE and LSTM-3 errors increase to 0.596% and 0.787%, respectively, whereas the PC-Neural-ODE error is still limited to 0.365%. The purely data-driven Neural-ODE and the LSTM-3 show reasonable accuracy but do not match the consistency of the PC-Neural-ODE which demonstrates its robustness under unseen operating conditions. Validation simulation 5 combines both scenarios, an extended prediction horizon (20000 seconds) and unseen operating conditions used in validation simulation 4. In this simulation, the Neural-ODE error increases to 0.876% with $R^2 = 0.995$ and the LSTM-3 error increases to 0.914% with $R^2 = 0.994$,

which is higher than the previous four validation cases. In contrast, the PC-Neural-ODE maintains a significantly lower error of 0.452% with $R^2 = 0.999$. This confirms the strong stability and generalization capability of the PC-Neural-ODE model. In the last validation simulation, the variation range of the inputs is extended until the PC-Neural-ODE shows a perceptible drop in accuracy (the fuel flow rate is varied between $10g/s$ and $90g/s$ and the water flow rate is varied between $0.6kg/s$ and $1.5kg/s$). The data driven Neural-ODE and LSTM-3 exhibit errors closer to 4%, whereas the PC-Neural-ODE achieves an error of 1.148%, which corresponds to a fourfold improvement in accuracy. These validation results highlight the clear advantage of embedding known physical constraints into the model. The added physics not only constrains the predictions to remain physically consistent, but also enhances its stability and generalization, particularly under operating conditions of the system that were not represented in the training data.

Table 1: Prediction accuracy comparison.

Validation	Neural-ODE vs 1D		PC-Neural-ODE vs 1D		LSTM-1 vs 1D		LSTM-2 vs 1D		LSTM-3 vs 1D	
Simulation	Error %	R^2	Error %	R^2	Error %	R^2	Error %	R^2	Error %	R^2
1	0.301	0.999	0.279	0.999	1.558	0.982	0.656	0.997	0.438	0.995
2	0.361	0.999	0.301	0.999	1.604	0.979	0.685	0.996	0.478	0.998
3	0.395	0.998	0.331	0.999	1.795	0.975	0.765	0.996	0.533	0.998
4	0.596	0.997	0.365	0.999	2.361	0.931	0.941	0.991	0.787	0.996
5	0.876	0.995	0.452	0.999	3.246	0.929	1.352	0.987	0.914	0.994
6	3.846	0.914	1.148	0.988	9.746	0.859	5.698	0.889	3.983	0.908

Training data requirement

The training data requirement of the models are evaluated by varying the number of simulation datasets used during training. Both the purely data-driven Neural-ODE and the PC-Neural-ODE are trained using 12 simulation datasets, which are sufficient to achieve accurate and stable predictive performance. In contrast, the LSTM model is trained on progressively larger datasets; 12 (LSTM-1), 20 (LSTM-2) and 30 (LSTM-3) simulations to assess its dependency on volume of data. As seen in Table 1, LSTM models exhibit a stronger dependence on the quantity of training data. Indeed, LSTM-1 which is trained on the same 12 simulation datasets as the Neural-ODE and the PC-Neural-ODE has significantly less accuracy, highlighted by higher prediction errors across all validation cases. When the amount of data is increased to 20 simulations (LSTM-2), the prediction accuracy improves, and a further increase to 30 simulations (LSTM-3) leads to additional accuracy improvement with errors that are still higher than Neural-ODE and PC-Neural-ODE but closer. These results demonstrate the superior data efficiency of the Neural-ODE approaches which formulate the dynamic system in continuous time rather than discrete time. By modeling the underlying physics of the heat exchangers in a continuous-time differential equation framework, the Neural-ODEs capture the intrinsic evolution of the system states independent of specific sampling intervals. In contrast, the LSTM model works in discrete-time by learning step to step mappings that are inherently tied to the temporal resolution and training data quantity. Across all training data sizes considered, continuous-time models demonstrated more stable performance and lower generalization errors under limited data conditions. These results indicate the comparatively higher data efficiency of continuous-time based models and their robustness in scenarios where limited data are available for training.

Computational efficiency

Computational efficiency is assessed by analyzing both the training and inference time of the proposed models. In terms of training time, Neural-ODE models are generally slower to train compared to discrete-time models. This is because each forward pass of a Neural-ODE involves solving an ODE and computing adjoint sensitivities for gradient estimation, which adds additional computational overhead during training. This computational cost also depends on the complexity of the learned dynamics and the specified solver tolerance to ensure numerical accuracy. LSTM models, on the other hand, rely on simple matrix multiplications and element-wise operations at each discrete time step, making

each forward and backward pass computationally cheaper compared to Neural-ODEs. All models were trained on a Dell computer with an Intel® Xeon® Gold 5218R CPU operating at $2.10GHz$. The purely neural-ODE model required approximately 4.5 hours for training, while the PC-Neural-ODE took around 3.8 hours. The reduced training time for the PC-Neural-ODE can be explained by the embedded physical dynamics, which simplifies the dynamics that the NN must approximate. In contrast, the LSTM models took only 14 minutes to train. Inference time is compared for a time span of 6000 seconds and is presented in Table 2. The 1D CHP model took 12.51 seconds and is used as a reference to evaluate how much computational acceleration the proposed models achieved. The purely data-driven Neural-ODE and the PC-Neural-ODE models are 763 and 1270 times faster, respectively, whereas the LSTM model achieved 750 times acceleration. It is seen that the PC-Neural-ODE is computationally more efficient than the pure Neural-ODE. This is because the embedded physical dynamics of the heat exchangers into the model makes its numerical behavior smoother and less stiff, allowing the numerical integrator to take larger adaptive steps while preserving accuracy.
Overall, the continuous-time models demonstrate a strong capability in representing the underlying dynamic behavior of the system, achieving prediction accuracy comparable to the LSTM model while requiring less training data. Embedding known physical dynamics into the model further improves its stability, confirming the benefit of incorporating physical knowledge into the model. From a computational time perspective, the discrete-time LSTM models achieve faster training times due to their discrete structure, whereas the continuous time models required greater computational effort because of the numerical integration through the ODE solver. These findings emphasize the trade-off between data efficiency, generalization, and computational cost for training and inference.

Table 2: Computational time and acceleration comparison.

	1D	Neural-ODE	PC-Neural-ODE	LSTM
Computational Time (s)	12.51	0.0164	0.00984	0.0167
Acceleration	-(ref)	763x	1270x	750x

5. Conclusion

Continuous and discrete-time machine learning approaches are evaluated to model the dynamic behavior of heat exchangers within a CHP system comprising three interconnected heat exchangers. Two continuous-time models, one purely data driven and the other with known physical dynamics embedded in it, are compared against a discrete-time LSTM model. Performance is compared in terms of prediction accuracy, training data requirement, generalization, and computational efficiency. The continuous-time models demonstrate superior performance in capturing the dynamics of the system with fewer training data. They also provide better generalization to unseen data, particularly the physics-constrained one, which closely aligns with the known dynamic behavior of the system. In contrast, the LSTM model needs approximately three times more training data to achieve comparable accuracy. From a computational time perspective, the continuous-time models have a higher training cost than the LSTM because of the numerical integration during training. In contrast they achieve higher acceleration during inference. Overall, the comparison highlights a trade-off between data efficiency, generalization, and computational requirements.

Acknowledgments

This work has benefited from state aid managed by the Agence Nationale de la Recherche under the France 2030 programme, reference 22-PETA-0002.

Nomenclature

Symbols

A	Area, m^2
C_p	Specific heat capacity, $J/(kg\,K)$
D_s	Shaft diameter, m
h	Heat transfer coefficient, $W/(m^2\,K)$
k	Specific heat ratio , $-$
M	Mass, kg
$\dot{m}$	Mass flow rate, kg/s
N	Shaft speed, rps
P	Power, W
r	Pressure ratio, $-$
T	Temperature, K
u	Velocity, m/s
$\dot{W}$	Rate of work, W

Greek symbols

η	Efficiency
ρ	Density, kg/m^3
ϕ	Flow coefficient

Subscripts and superscripts

a	Air
c	Cold
cc	Combustion chamber
co	Compressor
fg	Flue gas
h	Hot
in	Inlet
o	Outlet
t	Turbine
w	Wall

References

[1] Al-Dawery SK., Alrahawi AM., Al-Zobai KM., *Dynamic modeling and control of plate heat exchanger*. International Journal of Heat and Mass Transfer 2012;55(23-24):6873-6880.

[2] Bastida H., Ugalde-Loo CE., Abeysekera M., Qadrdan M., *Dynamic modeling and control of a plate heat exchanger*. In: 2017 IEEE conference on energy internet and energy system integration (EI2) 2017 Nov 26: 1-6. IEEE.

[3] Rasouli E., Tano IN., Aboud A., Seo J., Lamprinakos N., Rollett A., Narayanan V., *Experimental characterization of an additively manufactured heat exchanger for high temperature and pressure applications*. Applied Thermal Engineering. 2025 Apr 1;264:125412.

[4] Singh A., Sahu D., Verma OP., *Study on performance of working model of heat exchangers*. Materials Today: Proceedings 2023;80:8-13.

[5] Momeni M., Fartaj A., *Numerical thermal performance analysis of a PCM-to-air and liquid heat exchanger implementing latent heat thermal energy storage*. Journal of Energy Storage. 2023 Feb 1;58:106363.

[6] Ward O., Galvanin F., Jurado N., Clements C., Abdallah M., Blackburn D., Fraga E., *Lumped-parameter heat exchanger models for the robust dynamic modelling of power generation cycles*. In:Computer Aided Chemical Engineering 2023 Jan 1;52:3271-3276.

[7] Desideri A., Dechesne B., Wronski J., Van den Broek M., Gusev S., Lemort V., Quoilin S., *Comparison of moving boundary and finite-volume heat exchanger models in the modelica language*. Energies. 2016 May 5;9(5):339.

[8] Gupta S., Gupta R., Padhee S., *Parametric system identification and robust controller design for liquid–liquid heat exchanger system*. IET Control Theory and Applications. 2018 Jul;12(10):1474-82.

[9] Laszczyk P., *Simplified modeling of liquid-liquid heat exchangers for use in control systems*. Applied Thermal Engineering. 2017 Jun 5;119:140-55.

[10] Altes-Buch Q., Dickes R., Desideri A., Lemort V., Quoilin S., *Dynamic modeling of thermal*

systems using a semi-empirical approach and the ThermoCycle Modelica Library. In28th International Conference on Efficiency, Costs, Optimization and Simulation of Energy Systems 2015.

[11] Zavala-Río A., Santiesteban-Cos R., *Reliable compartmental models for double-pipe heat exchangers: An analytical study*. Applied Mathematical Modelling. 2007 Sep 1;31(9):1739-52.

[12] Hou G., Zhang D., An Z., Yan Q., Jiang M., Wang S., Ma L., *Evaluating high-precision machine learning techniques for optimizing plate heat exchangers' performance*. Energies. 2025 Feb 17;18(4):957.

[13] Zou J., Hirokawa T., An J.., Huang L., Camm J., *Recent advances in the applications of machine learning methods for heat exchanger modeling—a review*. Frontiers in Energy Research. 2023 Nov 14;11:1294531.

[14] de Vicente VS., da Silva LR., Scariot MR., *Machine learning for predicting the temperature profile of heat exchanger*. Studies in Engineering and Exact Sciences. 2025 Mar 21;6(1):e14966-.

[15] Santiago Galicia E., Hernandez-Matamoros A., Miyara A., *Machine learning-driven prediction of heat transfer coefficients for pure refrigerants in diverse heat exchangers types*. Journal of Experimental and Theoretical Analyses. 2025 Oct 16;3(4):32.

[16] Çolak AB., Açıkgöz Ö., Mercan H., Dalkılıç AS., Wongwises S., *Prediction of heat transfer coefficient, pressure drop, and overall cost of double-pipe heat exchangers using the artificial neural network*. Case Studies in Thermal Engineering. 2022 Nov 1;39:102391.

[17] Hughes MT., Chen SM., Garimella S., *Machine-learning-based heat transfer and pressure drop model for internal flow condensation of binary mixtures*. International Journal of Heat and Mass Transfer. 2022 Sep 15;194:123109.

[18] Hosseini S., Khandakar A., Chowdhury ME., Ayari MA., Rahman T., Chowdhury MH., Vaferi B., *Novel and robust machine learning approach for estimating the fouling factor in heat exchangers*. Energy Reports. 2022 Nov 1;8:8767-76.

[19] Zhang X., Zou Y., Li S., Xu S., *A weighted auto regressive LSTM based approach for chemical processes modeling*. Neurocomputing. 2019 Nov 20;367:64-74.

[20] Hochreiter S., Schmidhuber J., *Long short-term memory. Neural computation*. 1997 Nov 15;9(8):1735-80.

[21] Chen ZY., Wang RY., Jiang R., Chen T., *Neural ordinary differential gray algorithm to forecasting nonlinear systems*. Advances in Engineering Software. 2022 Nov 1;173:103199.

[22] Yang Y., Li H., *Neural ordinary differential equations for robust parameter estimation in dynamic systems with physical priors*. Applied Soft Computing. 2025 Jan 1;169:112649.

[23] Zhu A., Jin P., Zhu B., Tang Y., *On numerical integration in neural ordinary differential equations*. In International Conference on Machine Learning 2022 Jun 28 (pp. 27527-27547). PMLR.

[24] Barsali S., De Marco A., Giglioli R., Ludovici G., Possenti A., *Dynamic modelling of biomass power plant using micro gas turbine. Renewable Energy*. 2015 Aug 1;80:806-18.

[25] Razmi AR., Afshar HH., Pourahmadiyan A., Torabi MJ., *Investigation of a combined heat and power (CHP) system based on biomass and compressed air energy storage (CAES)*. Sustainable Energy Technologies and Assessments. 2021 Aug 1;46:101253.