Evaluation of Graph Neural Networks as Surrogate Model for District Heating Networks Simulation

***Taha Boussaid** [a, b, *], **François Rousset** [a], **Vasile-Marian Scuturici** [b], **Marc Clausse** [a]*

[a] Univ. Lyon, INSA Lyon, CNRS, CETHIL, UMR 5008, F-69621 Villeurbanne, France
[b] Univ. Lyon, INSA Lyon, CNRS, LIRIS, UMR 5205, F-69621 Villeurbanne, France
[*] Corresponding Author, taha.boussaid@insa-lyon.fr

Abstract:

District heating networks have proven their higher conversion efficiency, economic viability and environmental benefits when compared to decentralized and individual heating systems. These benefits are achieved through the ability to incorporate a wide variety of production means, including renewable intermittent sources but also via the use of short-term and/or inter-seasonal storage.

Due to the numerous interactions between these components, their different dynamic aspects and operating constraints, physical simulations are computationally heavy so that running optimization tasks become prohibitively expensive and time consuming. Therefore, new control optimization schemes need to be drawn up to accelerate the predictive control and to facilitate the decision-making process.

In the present work, we assess the application of geometric deep learning as a surrogate modeling framework for district heating simulations. Beyond processing non-Euclidian data, this deep learning approach aims to encode geometric and topological understandings of data as inductive biases in deep learning models. More precisely we trained Graph Neural Networks to emulate a thermo-hydraulic simulator of district heating network. This statistical inference method allows us to drastically reduce simulation time, hence unlocking further optimization loops and parametric space exploration. In addition, their permutation equivariance and stability to perturbations are assessed to discuss their scalability to more complex network topologies and control schemes.

Keywords:

District heating networks simulation, graph neural networks, transient dynamics.

Nomenclature

Letter symbols

c	specific heat, $\mathrm{J/(kgK)}$
T	temperature, K
$\dot{m}$	mass flow rate, $\mathrm{kg/s}$
N	number of nodes
N_b	number of branches
M	incidence matrix, $\in \mathbb{R}^{N \times N_b}$
KS	area independent heat transfer coefficient of pipe, $\mathrm{W/K}$
V	volume of the fluid contained in the control volume, m^3
F_ω	inference function

Greek symbols

$\gamma,\ \phi$	differentiable functions
ω	learnable parameters (weights) of inference function
σ	non-linear activation function

Subscripts and superscripts

i	node i
l	at layer l
t	time-step or learning step
s	relative to soil
b	relative to branches
ext	relative to exterior

https://doi.org/10.52202/069564-0286

1. Introduction

Anthropic activities have increased net greenhouse gas emissions since 2010 in all major sectors (industry, energy, transport, agriculture, construction). A study by IPCC Group III [1] investigated ways to mitigate climate change and reduce greenhouse gas emissions. According to their report, switching from fossil fuels to low-carbon energies is essential to limit climate change. In Europe, the production of heat and cold represents half of the energy consumption and is mainly based on fossil fuels [2]. Since the last century, district heating networks (DHN) have been deployed mainly due to their economical and efficiency benefits. For example, having a joint production process, a higher conversion rate and less maintenance allows to decrease the operating and maintenance costs. Moreover, DHN are well suited for areas of mixed use with strong anchor clients [3]. In recent years, due to the rapid development of renewable energy technologies, the 4th generation of DHN have been identified as a viable option to decarbonize the heat production sector [4, 5]. First, they operate at lower temperatures, hence reducing heat losses and increasing the efficiency of conversion systems. They also take advantage of various heat sources including renewable and recycled ones such as biomass, geothermal, solar thermal, and waste heat. Finally, coupling these sources with thermal energy storage (TES) and optimal control should allow a better peak management. However, the use of TES and various heat sources, some of which are intermittent, adds a new complexity to the system including the stochastic character of some variables such as resources availability, weather conditions and electricity price, which therefore implicitly integrate a predictive character in the control and optimization methods of these systems.

In order to successfully operate these advanced thermal networks, an intelligent control strategy is required. A detailed review on control strategies for DHN can be found in [6]. This study shows that the current tendency is to use hybrid control schemes based on multi-agent systems and model predictive control (MPC). This method falls in two parts: first, solving an operational optimization centrally i.e. energy planning, and then it is distributed to other decentralized agents related to consumers and the different producers connected to the network. This strategy involves the simulation of the dynamic behavior of the considered DHN. In fact, the controller needs to understand and predict the behavior of the system and its response to various scenarios of control variables which requires multiple simulations over variable large time horizons. To accomplish this, accurate models of the network's different components and their interactions are required. Several studies have been carried out to develop numerical models of DHN [7–11]. They can be classified in two main categories: dynamic and pseudo-dynamic models. The latter approach considers only the thermal transient and assumes instantaneous hydraulic changes as fluid dynamic perturbations (i.e. pressure variations) are quickly transferred to the whole network, about 1000 times faster than temperature variations [8]. More importantly, when considering intermittent heat sources like solar thermal, it is crucial to use an adapted small time-step for the simulations. Consequently, this makes running iterative optimization methods computationally heavy and time consuming.

One solution to overcome this limitation is the formulation of a numerically efficient and stable surrogate model of DHN simulations. There are several surrogate modeling approaches [12] such as reduced order models, Gaussian kriging, radial basis functions, etc. But lately and due to extensive research and advances in computing technologies, different studies investigated the application of machine learning algorithms as surrogate models for different engineering systems including energy conversion and distribution processes [13–17]. For example, the study in [16] consisted on training a graph neural network to approximate the optimal solution of power flow optimization problem for electric grids. The results showed that the surrogate model is 10^5 faster than the interior point optimization method and it exhibits much better scalability to larger networks. A tutorial review on using neural network for MPC is given in [17]. More specifically, it highlights the use of recurrent neural networks (RNN) to predict the state of the process model. The authors applied this approach to drive the reactants concentration of a non-isothermal continuous stirred-tank reactor (CSTR) to the optimal steady-state point by controlling heating rate and inlet concentration. It was found that the physics based RNN-MPC system converges faster and requires less computational time.

In the field of DHN design optimization and control, machine learning was applied at two different levels. The first one focuses on predicting the thermal load [18, 19] and the second, more recent, on applying deep reinforcement learning to train autonomous agents to optimal DHN operation [20]. However, to the best author's knowledge, no attempt to formulate a surrogate model of DHN simulation using machine learning have been made. Therefore the primary contributions of this work are as follows:

- A flexible physical pseudo-dynamic simulator of district heating network has been developed based on graph theory.
- To the best author's knowledge, this is the first attempt to apply GNN as surrogate model for physical district heating network transients.
- This approach is validated through the prediction of district heating network behavior as the temperatures evolution in each node of the network.
- Using this framework allows to significantly reduce simulation time from hours to seconds, $\sim 10^4$ faster.

https://doi.org/10.52202/069564-0286

This opens the possibility to further develop quasi instantaneous optimal control schemes.

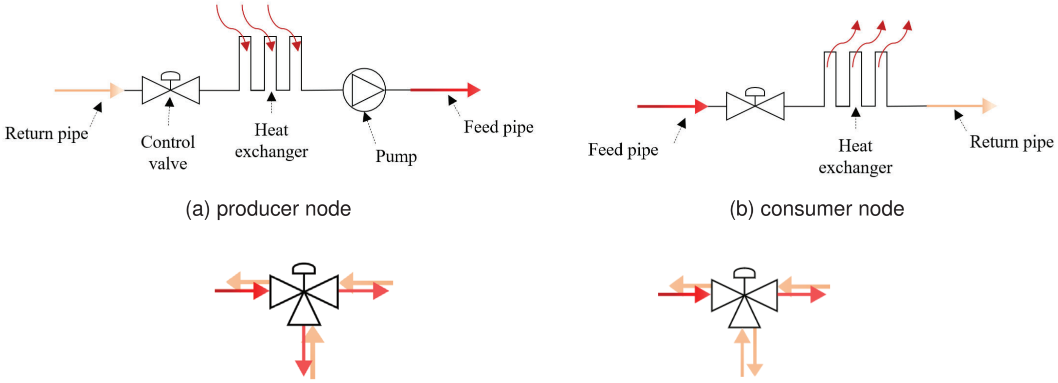
The remaining of the paper is organised as follows: Section 2. describes the physical model that has been developed to simulate transient dynamics in DHN. Section 3. briefly introduces fundamentals of machine learning with a focus on GNN and the final architecture of our surrogate model. Then, section 4. presents the application case that was chosen to evaluate our approach. Finally, the results, limitations and future developments are presented in section 5.

2. Physical model

The physical model that has been developed is intended to serve as a data generator for DHN simulations similarly to what has been done on the electrical network side with the MATPOWER test cases project [21]. The choice of pseudo-dynamic resolution is justified as the time-step of DHN simulations is generally around one hour and recently minutes with the development of IoT sensors. The hydraulic changes are in fact quick and are transferred to the whole network in a period of time of seconds.
The model has been developed in `Python` language. We used `TESPy` (Thermal Engineering Systems in Python) package [22], an open-source physical solver previously validated on different use cases [23–25]. To simulate steady and transient dynamics of DHN, a customised code layer has been built upon this physical solver. Mainly, the first step consisted on defining the topology of the network using graph theory, then physical properties were assigned to each component in order to run the simulations.

2.1. Graph representation

The topology of each DHN is defined using graph theory. Let $\mathcal{G} = (V, E)$ be a graph with a set of N nodes V and a set of N_b edges E. Physically, the edges are the pipes of the network, while the nodes can be either consumers, producers or distribution/control valves. As the flow direction is an important variable, DHN are represented as directed graphs. Therefore, the simulator needs two variables to completely define the DHN. The first is the *oriented incidence matrix* $M \in \mathbb{R}^{N \times N_b}$, which has a row for each node i and a column for each edge j such as $m_{ij} = 1$ if edge j enters the vertex i, $m_{ij} = -1$ if the edge leaves the node and $m_{ij} = 0$ if the two elements are not connected. The second variable is the *nodes vector* $V_0 \in \mathbb{R}^N$ that defines the type of each node with regard to a predefined set of supported node types schematized in Fig.1.
Currently, the consumer nodes are considered to be heat sinks without modeling the sub-station part, producer nodes can be one of the heat sources implemented at this time: gas boiler, biomass boiler and solar thermal collectors. Finally, two types of distribution valves are modeled, 3 ways valves and cascading consumers valve. The next step is to assign the suitable physical characteristics to each component. The methodology and the physical models implemented are detailed in appendix A.



(a) producer node

(b) consumer node

(c) 3-ways valve

(d) cascading valve

Figure 1: The components of each node type supported by the physical simulator

2.2. Numerical methods

Following the graph representation of DHN, the simulator solves the corresponding thermo-hydraulic equations based on the pseudo-dynamic regime assumption. Via a control volume approach, simulating transient dynamics of DHN yields to the following equations for the hydraulic part (see [9]):

$$M \cdot G + G_{ext} = 0, \tag{1}$$

$$M^\intercal \cdot P = R \cdot G - W. \tag{2}$$

In Eq.1, $G \in \mathbb{R}^{N_b}$ is the vector containing the flow of each branch, $G_{ext} \in \mathbb{R}^{N}$ the vector containing the extractions or injections of flows that occur at the nodes.

In Eq.2 $R \in \mathbb{R}^{N_b \times N_b}$ is a diagonal matrix containing the hydraulic resistance of each pipe, $W \in \mathbb{R}^{N_b}$ contains the pressure increases due to the work of the pumps and $P \in \mathbb{R}^{N}$ the vector containing the values of the *total pressure* in each node. It should be noted that equations 1 and 2 form a non-linear system given that R depends on G. This system is solved using Newton-Raphson algorithm, G and P being the unknowns of the system.

After the steady hydraulic regime is solved, i.e. the pressure and mass flow rate in each node are known, the thermal transient regime is obtained by following the direction of flow in the network and considering each pipe as a control volume with the outlet temperature as unknown (Fig.8a):

$$\rho c_p V_j \frac{dT_i}{dt} = c_p G_j \left(T_{i-1} - T_i\right) - K S_j \left(\frac{T_{i-1} + T_i}{2} - T_s\right) \tag{3}$$

In case of more than one pipe entering a node i, the temperature of this node is calculated considering a perfect mixture assumption. The heat losses are calculated considering the average temperature of each pipe, this assumption is generally applied in numerical models considering that the temperature drops very slightly along a single pipe segment between two nodes [27]. Formulating Eq.3 for the whole network leads to a linear system that is solved using Euler implicit scheme.

3. Surrogate model

In this section we introduce the surrogate model (SM) architecture that was used for our imitation learning problem. The output of the model can be any variable of interest, but without loss of generality, in this work the SM is trained to predict the evolution of the return temperature at each node given the network topology, its physical characteristics, heat demand and a set of control laws.

3.1. Supervised learning

The aim is to train the SM to emulate the thermo-hydraulic simulator of a DHN. More precisely, let X be the set of inputs given to the physical model and Y the outputs through the physical equations detailed in section 2.2. The SM takes the same inputs and imitate the physical simulator by constructing an *inference function* F_ω such that $Y_\omega = F_\omega\left(X\right)$ where ω is a vector of parameters, called weights, that are adjusted during the training step to minimize a specific loss function $\mathcal{L}_\omega$ (i.e. residual function), for example the L_2 norm (i.e. Euclidean norm) $\mathcal{L}_\omega = \|Y_\omega - Y\|_2$.

This is known as a supervised learning problem since we have both the input and output data, unlike unsupervised learning where the goal of the model is to find patterns in the input data to classify them for example. More schematically, the learning process corresponds to an optimization loop where the weights of the model are corrected iteratively in order to reach a global minimum of the loss function as illustrated in Fig.2. The figure also introduces the first *hyper-parameter* of the SM: the learning rate α which modulates the convergence speed of the model.

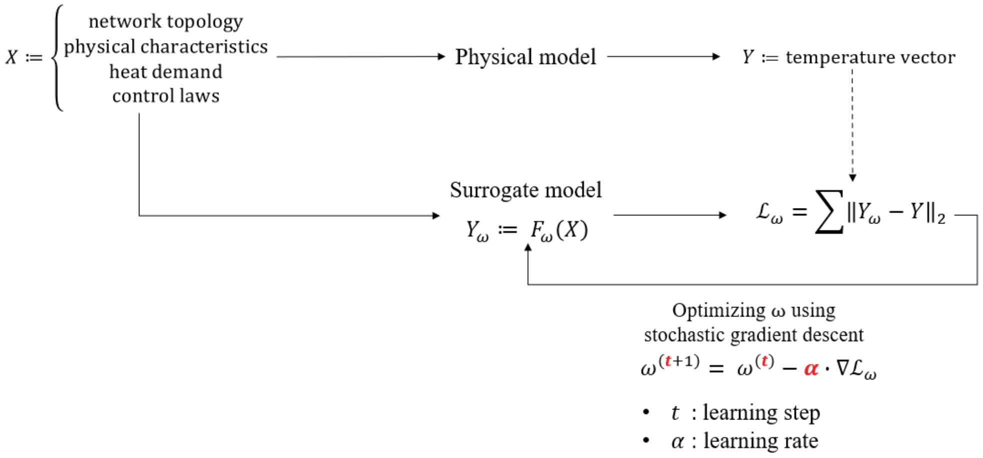

$X :=$
network topology
physical characteristics
heat demand
control laws
Physical model
$Y :=$ temperature vector
Surrogate model
$Y_\omega := F_\omega(X)$
$\mathcal{L}_\omega = \sum \|Y_\omega - Y\|_2$
Optimizing ω using
stochastic gradient descent
$\omega^{(t+1)} = \omega^{(t)} - \alpha \cdot \nabla \mathcal{L}_\omega$
• t : learning step
• α : learning rate

Figure 2: Supervised learning loop of a surrogate model

https://doi.org/10.52202/069564-0286

3.2. Multi-Layer perception networks

The task of constructing the inference function F_ω is an active research area in machine learning. However, deep learning is a particular sub-domain of machine learning where extensive developments are being made continuously. This framework is mainly based on artificial neural networks (ANN) as the learning structure. The simplest ANN is called Multi-Layer Perceptron (MLP) which consists of interconnected layers of artificial neurons, named perceptrons. The neurons in each layer are connected to the neurons in the next layer, forming a directed acyclic graph. Each neuron receives inputs from the neurons in the previous layer, applies a linear transformation to the inputs, and passes the result through a non-linear activation function. The output of a neuron i in layer l is computed as follows:

$$x_i^{(l+1)} = \sigma\left(\sum_{j=1}^{n_l} w_{ij}^{(l+1)} x_j^{(l)} + b_i^{(l+1)}\right), \tag{4}$$

where $x_j^{(l+1)}$ is the output of neuron j in the previous layer, $w_{ij}^{(l+1)}$ is the weight of the connection from neuron j in layer (l) to neuron i in layer $(l+1)$, $b_i^{(l+1)}$ is the bias of neuron i in layer $(l+1)$, n_l is the number of neurons in layer (l) and σ is a non-linear activation function, for example the rectified linear unit function $\texttt{ReLU}(t) = \max(0, t)$. Intuitevely, the activation function either allows the neurons to communicate its output to the next one (i.e. activated) or not when the output does not meet the activation criterion.

Finally, the output of the MLP is the output of the last layer of neurons $\hat{y} = x^{(L)}$. Where $\hat{y}$ is the predicted output, and L is the number of layers in the MLP.

3.3. Graph neural networks

Message passing operation

GNN are a class of neural networks designed to operate on data represented as graphs. GNN aim to learn representations of nodes or the entire graph, which can be used for various tasks such as node classification, node regression, link prediction, etc. The key idea behind GNNs is to iteratively update the representation of each node by aggregating information from its neighboring nodes. This is done by *passing messages* between nodes using a set of learnable parameters. The message passing operation can be formulated as in [28]:

$$\mathbf{x}_i^{(l+1)} = \gamma_\omega\left(\mathbf{x}_i^{(l)}, \bigoplus_{j\in\mathcal{N}(i)} \phi_\omega\left(\mathbf{x}_i^{(l)}, \mathbf{x}_j^{(l)}, \mathbf{e}_{j,i}^{(l)}\right)\right), \tag{5}$$

where $x_i^{(l)}$ is the representation of node i at layer l, $\bigoplus$ a differentiable permutation invariant function (sum, mean, etc.) and $\mathcal{N}(i)$ is the set of neighboring nodes of node i. Finally, γ_ω and ϕ_ω are differentiable functions with learnable weights ω such as MLPs defined in section 3.2. $\mathbf{e}_{j,i}$ is the edge representation connecting node j to node i.

In each layer of a GNN, the representations of all nodes are updated using the message passing operation, resulting in a new set of node representations at the next layer. This process is repeated for multiple layers, allowing the GNN to capture increasingly complex dependencies between nodes. $X^{(l)} = [x_1, \dots x_N]^\intercal$ is the matrix containing the updated representations of the nodes at layer l. If the GNN has L layers, then one has $X^{(0)} = X$ i.e. the initial inputs of the model and $X^{(L)} = Y_\omega$ i.e. the final outputs. For a more guided and detailed presentation on GNN, see [28–30].

Graph attention networks

Graph Attention Networks (GATs) are a type of GNN that uses attention mechanisms to selectively aggregate information from neighboring nodes [31]. The attention mechanism allows GATs to learn different weights for different neighbors of each node, depending on their relevance to the prediction task. Following the definition in Eq.5, the graph attention operation is written as:

$$\mathbf{x}_i^{(l+1)} = \sigma\left(\alpha_{i,i}\mathbf{W}\mathbf{x}_i^{(l)} + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\mathbf{W}\mathbf{x}_j^{(l)}\right), \tag{6}$$

where $\alpha_{i,j}$ is the attention coefficient for each neighbor j of node i and σ a non-linear activation function. In a GAT, the representation of each node i at layer l is computed as a *weighted sum* of its neighbors' representations, where the weights are learned using an attention mechanism [31]. The interest of this method here is that physically, nodes in a DHN are first impacted by their upstream neighboring node along the direction of flow, and then by the other nodes in a lesser degree.

4. Study case

4.1. Model architecture

The inference problem shown in Fig.2 involves the use of heterogeneous inputs. Here we will distinguish them according to their static or dynamic character and then according to whether these data are local or global, in other terms whether they are defined at nodes and edges levels or at graph level. First, let $\mathcal{H}$ be the horizon of prediction of the SM and Δt the time-step such that $\mathcal{H} = p \times \Delta t$ and p the number of data points. We define:

- Static nodes attributes: here we simply map the node type to a predefined scalar $\{producers = 0,\ valves = 0.5,\ consumers = 1\}$. This input will be noted $X_s \in \mathbb{R}^N$
- Static edges attributes: as each edge represents a pipe, the attributes are the pipe length L, inner diameter D and its equivalent thermal resistance KS. Subsequently, this variable is noted $E_s \in \mathbb{R}^{N_b \times 3}$.
- Dynamic nodes attributes: it consists of a matrix containing the evolution of heat demand at each node that will be noted $X_d \in \mathbb{R}^{N \times p}$. For distribution valves and producers, the heat demand is computed as the sum of the heat load at their children nodes.
- Global dynamic attributes: this input includes the control signals that affect the whole network, in our case these will be the evolution of the supply temperature $T_s \in \mathbb{R}^p$ and total mass flow rate $G_{tot} \in \mathbb{R}^p$. This input also holds the evolution of the external temperature along the prediction horizon $T_{ext} \in \mathbb{R}^p$.

The network topology is implicitly included during the learning phase via the *adjacency* list that is used by the graph attention operation in order to select the neighbors of each node. Following the above definitions, the SM architecture is in the form of an encoder-processor-decoder as shown in Fig.3. The encoder consists of two MLPs that transform the input data and project it into a latent space where each node is assigned two hidden vectors representing respectively its local attributes (heat demand) and also information from global variables updated with neighbors representations using three graph multi-head attention layers. The processor parts consists on transforming the node hidden representations using three multi-headed attention operations, found to be more efficient than single headed attention in [31]. Finally the decoder is composed of two linear layers that maps the nodes representation to the output space. The model was implement using `Pytorch geometric` library [33].

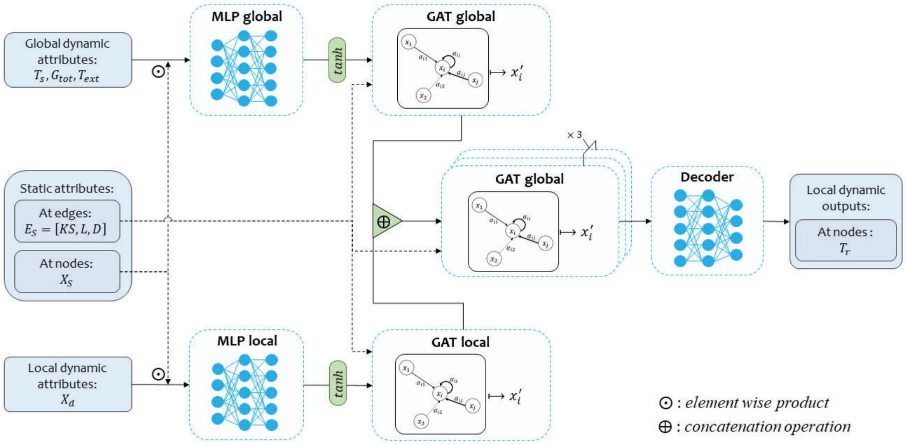


Figure 3: Surrogate model encoder-processor-decoder architecture

4.2. Data generation

In order to assess our approach, we chose to apply it on the academic study case modeled in [32]. This choice is based on the availability of data needed by the physical simulator. The site includes buildings of different usage profiles as shown in Fig.4 where the number under each node indicates its index. Given the annual heat load for each consumer and a typical external temperature variation, we generate an hourly consumption profile for each building using the geometric series approach implemented in `demandlib` library [34], a Python package that allows to create heat profiles from annual values. The supply temperature T_s depends on the outside temperature according to the adapted water law from [35]. For the total flow rate G_{tot}, it is determined

https://doi.org/10.52202/069564-0286

by energy balance over the entire network. The network operation is then simulated for one year with a time step $\Delta t = 600s$. The simulation time is equal to $t_{sim,4c} = 182.1min$ using a four cores CPU processor, which is equivalent to $t_{sim,1c} = 636.2min$ of single core CPU time.
The outputs are time series of $8760 \times 6 = 52560$ data points for each node, representing the evolution of temperatures. Then, we split the data into two datasets, 75% for training and 25% for validation. The horizon of prediction defined in section 4.1. is set to 1 week $= 7 \times 24 \times 6 = 1008$ data points. Thus 39 weeks are used for training and 13 for the validation. Typical values were used for the model hyper-parameters:

- The learning rate α is set to $\alpha_0 = 1 \cdot 10^{-3}$ with an exponential decay factor of $k = 5 \cdot 10^{-4}$, meaning that $\alpha = \alpha_0 \times e^{-kt}$ where t is the epoch number. Therefore, the learning rate is reduced as the training advances which slows down the gradient descent as the loss function approaches its minimum.
- The batch size is the number of samples processed before the model is updated. Here it is set to 8, a compromise between the volume of available data and a large batch size.
- The number of epochs is the number of complete passes through the training dataset. In our case, the model was trained over 200 epochs.
- Finally, L_2 norm was used as the loss function.

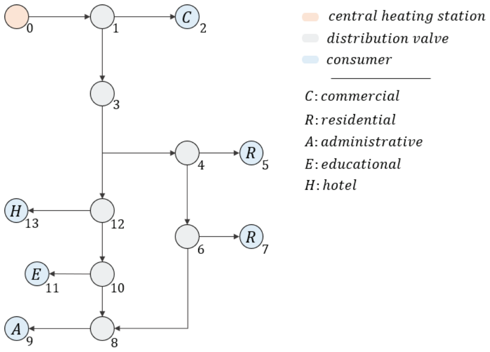


Figure 4: DHN topology of the study case adapted from [32]

5. Results and discussion

After the training phase, the model has been tested on the validation dataset. The root mean squared error (RMSE) is used to evaluate the model performances. It can be seen from table 1 that the error stays below 0.7K for all nodes. Moreover, we report a more visual plot in Fig.5a & 5b for predictions of the node 9, which is an administration building, and the furthest consumer node from the central heating plant. Also, the predictions for node 13, which is a different consumer type, are given in Fig.5c & 5d. More precisely, we plot the normalized temperature (Eq.7), this scaled variable is inherent to the the SM training that fits better with normalized variables (both inputs and outputs) which helps to stabilize the gradient descent step.

$$T^* = \frac{T - T_{min}}{T_{max} - T_{min}}, \tag{7}$$

It can be seen that the prediction results are very close to the values obtained from the physical simulation.

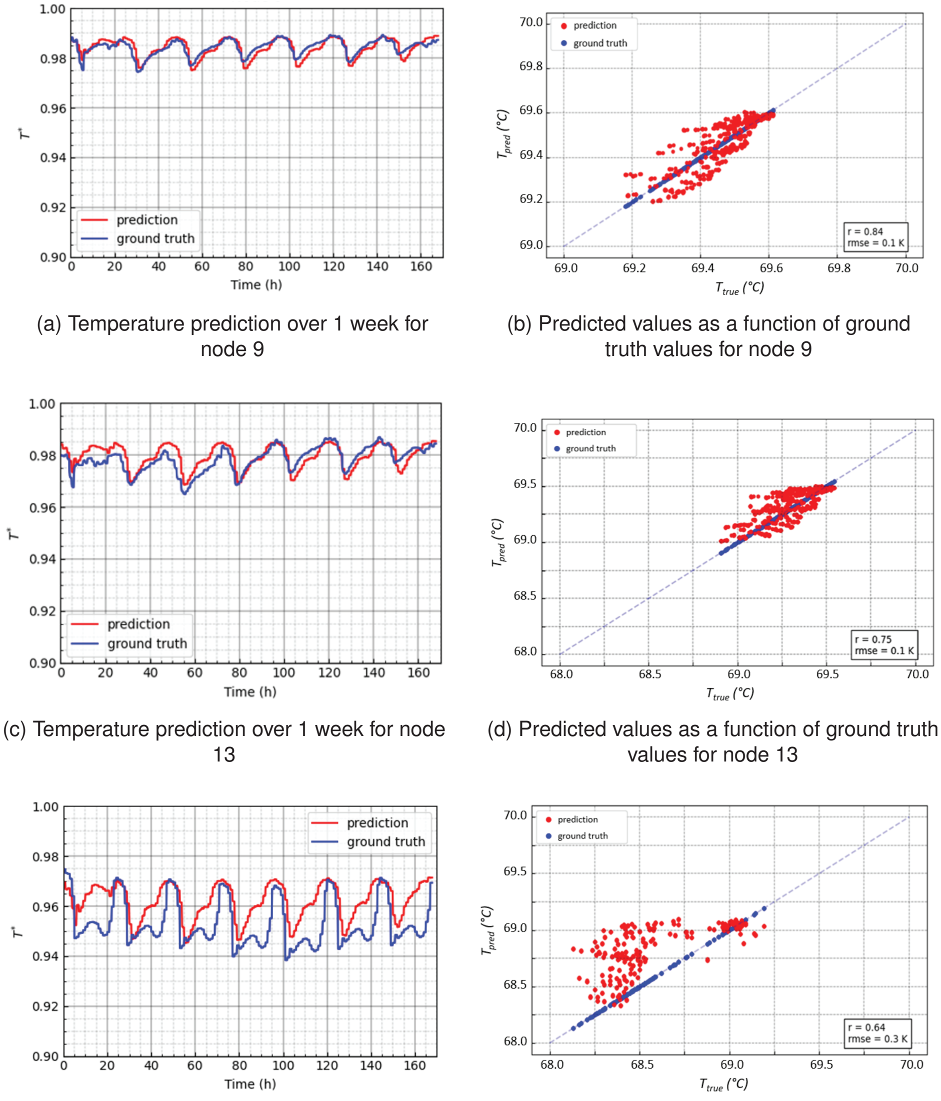
Table 1: RMSE for return temperatures prediction on validation dataset

node	0	1	2	3	4	5	6	7	8	9	10	11	12	13
RMSE (K)	0.5	0.3	0.7	0.2	0.3	0.3	0.5	0.3	0.2	0.1	0.1	0.1	0.1	0.1

The biggest deviations are found at the peaks where the SM underestimates the minimum temperature values and slightly overestimates the maximum values. We also added the Pearson correlation coefficient r that is equal to 0.84 in this case, reflecting a good correlation. The inference time is equal to $t_{inference} = 0.15s$ against $t_{sim,4c} = 2700s$ for the physical simulation on the validation dataset which accounts for approximately 1.8×10^4 time gain.

https://doi.org/10.52202/069564-0286

The regularity observed in the return temperature is the direct effect of the heat profile of the consumer and also the small impact of the control variables that vary quite moderately. Additionally, the same results are shown for node 4 in Figs.5e & 5f. In this case, we observe that the trend is not perfectly fitted while the SM still captures the peak values. Given the scales on both figures, the error, RMSE $= 0.3K$ is still small and acceptable. However, this overestimation of temperature will impact the energy balance on the graph level. This indicates a lack of physical representation in the SM parameters.



(a) Temperature prediction over 1 week for node 9

(b) Predicted values as a function of ground truth values for node 9

(c) Temperature prediction over 1 week for node 13

(d) Predicted values as a function of ground truth values for node 13

(e) Temperature prediction over 1 week for node 4

(f) Predicted values as a function of ground truth values for node 4

Figure 5: Example of predictions for nodes 4 (valve), 9 (administration building) and 13 (hotel)

In order to test further the model adaptability and its sensitivity to control laws, we tested it with new simulations where the supply temperature curve and the total mass flow rate were changed more often. Physically, this implied that the return temperature at each node was more affected by the control variables than the heat load itself. The model was not able to capture the correct patterns as illustrated in Fig.6. This limitation originates from an imbalance between the two encoding blocks where the local MLP seems to affects more the prediction. One way of adressing this problem is to incorporate physical laws during the learning phase. More precisely, it is possible to add physical constraints on the predicted values as part of the loss function, in our case this could be the mass and energy conservation equations that have to be verified at each iteration. Another line of exploration is to use recurrent neural networks such as Gated Recurrent Units (GRU) or Long-Short Term Memory (LSTM) cells that better incorporate the notion of temporality and time dependence into the inference function. Finally, another option is to train the model on more data and different network structures which will allow the SM to better emulates the physical behaviour of DHN.

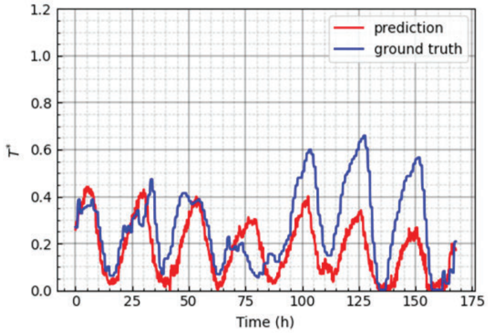


Figure 6: Example of prediction for node 13 (hotel) with a different control law

6. Conclusion

To conclude, the use of geometric deep learning as a surrogate modeling framework for district heating simulations has been evaluated in this study and offers promising results. By leveraging the ability of this approach to process non-Euclidean data and incorporate geometric and topological understandings of data into deep learning models as inductive biases, Graph Neural Networks were trained to emulate a thermo-hydraulic simulator of a district heating network. This statistical inference method has enabled us to significantly reduce simulation time by 10^4 orders of magnitude. Nevertheless, this approach was inadequate to properly capture the physical relations when the control laws are sharper. Therefore, we are currently working on incorporating physical constraints in the surrogate model to alleviate this limitation along with using recurrent structure to leverage more the time dependencies of the data. These improvements will be the subject of a future publication.

7. Acknowledgments

This work was supported by the French Ministry of Higher Education and Research.

https://doi.org/10.52202/069564-0286

References

[1] Masson-Delmotte, V., P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield *IPCC, Summary for Policymakers. In: Global Warming of 1.5℃. An IPCC Special Report on the impacts of global warming of 1.5℃ above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. Cambridge University Press 2018, Cambridge, UK and New York, NY, USA, pp. 3-24.

[2] Fleiter, T. *Profile of heating and cooling demand in 2015*. European Union's Horizon 2020; 2017.

[3] Delangle, Axelle and Lambert, Romain SC and Shah, Nilay and Acha, Salvador and Markides, Christos N. *Modelling and optimising the marginal expansion of an existing district heating network*. Energy 2017;140:209-223.

[4] Lund H, Werner S, Wiltshire R, Svendsen S, Thorsen JE, Hvelplund F, Mathiesen BV. *4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems*. Energy. 2014 Apr 15;68:1-1.

[5] Li H, Nord N. *Transition to the 4th generation district heating-possibilities, bottlenecks, and challenges*. Energy Procedia. 2018 Sep 1;149:483-98.

[6] Vandermeulen A, van der Heijde B, Helsen L. *Controlling district heating and cooling networks to unlock flexibility: A review*. Energy. 2018 May 15;151:103-15.

[7] Zheng X, Xu B, Wang Y, You S, Zhang H, Wei S, Wang N. *Hydraulic transient modeling and analysis of the district heating network*. Sustainable Energy, Grids and Networks. 2021 Mar 1;25:100409.

[8] Chertkov M, Novitsky NN. *Thermal transients in district heating systems*. Energy. 2019 Oct 1;184:22-33.

[9] del Hoyo Arce I, López SH, Perez SL, Rämä M, Klobut K, Febres JA. *Models for fast modelling of district heating and cooling networks*. Renewable and Sustainable Energy Reviews. 2018 Feb 1;82:1863-73.

[10] Van der Heijde B, Aertgeerts A, Helsen L. *Modelling steady-state thermal behaviour of double thermal network pipes*. International Journal of Thermal Sciences. 2017 Jul 1;117:316-27.

[11] Guelpa E, Sciacovelli A, Verda V. *Thermo-fluid dynamic model of large district heating networks for the analysis of primary energy savings*. Energy. 2019 Oct 1;184:34-44.

[12] Alizadeh R, Allen JK, Mistree F. *Managing computational complexity using surrogate models: a critical review*. Research in Engineering Design. 2020 Jul;31:275-98.

[13] Sun L, Liu T, Wang D, Huang C, Xie Y. *Deep learning method based on graph neural network for performance prediction of supercritical CO_2 power systems*. Applied Energy. 2022 Oct 15;324:119739.

[14] Fusco F, Eck B, Gormally R, Purcell M, Tirupathi S. *Knowledge-and data-driven services for energy systems using graph neural networks*. In 2020 IEEE International Conference on Big Data (Big Data) 2020 Dec 10 (pp. 1301-1308).

[15] Ryu S, Kim H, Kim SG, Jin K, Cho J, Park J. *Probabilistic deep learning model as a tool for supporting the fast simulation of a thermal–hydraulic code*. Expert Systems with Applications. 2022 Aug 15;200:116966.

[16] Owerko D, Gama F, Ribeiro A. *Optimal power flow using graph neural networks*. In ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020 May 4 (pp. 5930-5934).

[17] Ren YM, Alhajeri MS, Luo J, Chen S, Abdullah F, Wu Z, Christofides PD. *A tutorial review of neural network modeling approaches for model predictive control*. Computers & Chemical Engineering. 2022 Aug 13:107956.

[18] Xie J, Li H, Ma Z, Sun Q, Wallin F, Si Z, Guo J. *Analysis of key factors in heat demand prediction with neural networks*. Energy Procedia. 2017 May 1;105:2965-70.

[19] Westermann P, Welzel M, Evins R. *Using a deep temporal convolutional network as a building energy surrogate model that spans multiple climate zones*. Applied Energy. 2020 Nov 15;278:115563.

https://doi.org/10.52202/069564-0286

[20] Zhang B, Ghias AM, Chen Z. *A double-deck deep reinforcement learning-based energy dispatch strategy for an integrated electricity and district heating system embedded with thermal inertial and operational flexibility*. Energy Reports. 2022 Nov 1;8:15067-80.

[21] Birchfield AB, Xu T, Gegner KM, Shetye KS, Overbye TJ. *Grid structural characteristics as validation criteria for synthetic networks*. IEEE Transactions on power systems. 2016 Oct 28;32(4):3258-65.

[22] Witte F, Tuschy I. *Tespy: Thermal engineering systems in python*. Journal of Open Source Software. 2020 May 21;5(49):2178.

[23] Gasanzade F, Witte F, Tuschy I, Bauer S. *Integration of geological compressed air energy storage into future energy supply systems dominated by renewable power sources*. Energy Conversion and Management. 2023 Feb 1;277:116643.

[24] Chen C, Witte F, Tuschy I, Kolditz O, Shao H. *Parametric optimization and comparative study of an organic Rankine cycle power plant for two-phase geothermal sources*. Energy. 2022 Aug 1;252:123910.

[25] Chen X, Hao X. *Exergy analysis of a ground-coupled heat pump heating system with different terminals*. Entropy. 2015 Apr 17;17(4):2328-40.

[26] Wang H, Meng H, Zhu T. *New model for onsite heat loss state estimation of general district heating network with hourly measurements*. Energy conversion and management. 2018 Feb 1;157:71-85.

[27] Fang T, Lahdelma R. *State estimation of district heating network based on customer measurements*. Applied Thermal Engineering. 2014 Dec 5;73(1):1211-21.

[28] Bronstein MM, Bruna J, Cohen T, Veličković P. *Geometric deep learning: Grids, groups, graphs, geodesics, and gauges.* arXiv preprint arXiv:2104.13478. 2021 Apr 27.

[29] Sanchez-Lengeling B, Reif E, Pearce A, Wiltschko AB. *A gentle introduction to graph neural networks*. Distill. 2021 Sep 2;6(9):e33.

[30] Daigavane A, Ravindran B, Aggarwal G. *Understanding convolutions on graphs*. Distill. 2021 Sep 2;6(9):e32.

[31] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. *Graph attention networks*. arXiv preprint arXiv:1710.10903. 2017 Oct 30.

[32] Belfiore F. *District heating and cooling systems to integrate renewable energy in urban areas*. 8620 EPFL; 2021.

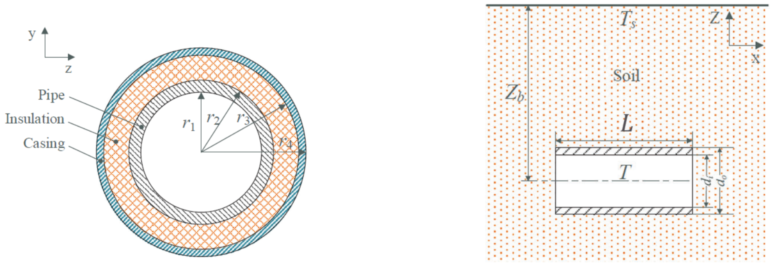
[33] Fey, M., Lenssen, J. E. *Fast Graph Representation Learning with PyTorch Geometric (2019)*. [Computer software]. `https://github.com/pyg-team/pytorch_geometric`

[34] Schachler B, Möller C et al. *demandlib: creating heat and power demand profiles from annual values*. `https://oemof.org/libraries/#demandlib`

[35] Kouhia M, Laukkanen T, Holmberg H, Ahtila P. *District heat network as a short-term energy storage*. Energy. 2019 Jun 15;177:293-303.

https://doi.org/10.52202/069564-0286

Appendix A Adding physical inputs to the simulator



(a) pipe cross section

(b) buried pipe

Figure 7: Schematic diagram of an insulated pipe [26]

Once the network topology defined, the physical properties of each component are fed to the model:

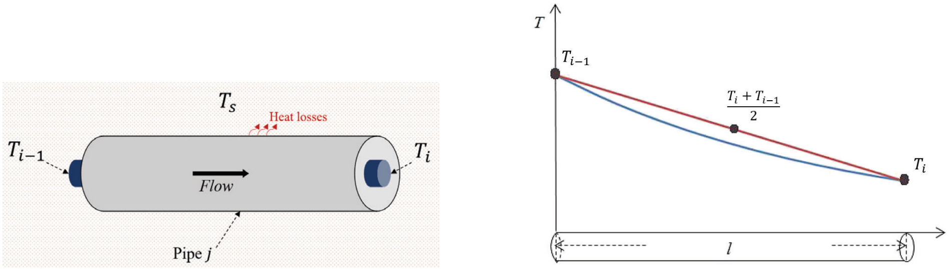
- Branches: the *oriented incidence matrix* M allows to create connections between nodes. Each edge corresponds to two pipes, the feeding pipe and the return one. We implement two layers insulated pipes as in [26]. Therefore, all the geometrical quantities $(L, r_1, r_2, Z_b \dots)$ shown in Fig.7 are inputs to the model, in addition to thermal conductivities k and roughness ϵ for determining pressure loss in the pipes.

- Producers: here, the inputs to the model vary depending on the heat source type. As the study case in section 4. considers only gas boiler, the required inputs are $\dot{Q}_{max}$ which is the maximum heat rate of the plant, pr the outlet/inlet pressure, and the isentropic η_s efficiency of the pump coupled to heat source.
- Valves: a quasi-incompressible fluid is assumed, which is always the case for liquids. The fluid flow between the inlet and the outlet of the valve is assumed to be isenthalpic. The flow rate is governed by the equation 8 derived from Bernoulli's law. K_ν is the required input by the model.

$$\dot{V} = K_\nu \sqrt{\Delta P}. \tag{8}$$

- Consumers: similarly the outlet/inlet pressure ratio is needed, and the heat load can be either a scalar if steady state resolution or a vector in the case dynamic simulation.

Finally, the model also needs global inputs that are the external and ground temperatures. The solver comes with its proper verification tools that make sure the system is nor over or under determined. Depending on the resolution scheme and the aim of the simulation, the unknowns of the system have to be chosen carefully. However, and without loss of generality, we will consider the particular case where the DHN is already designed, the variables of our system are the supply temperature at the heat source T_s and the total mass flow rate G_{tot}.



(a) control volume used in thermal transient equation

(b) pipe temperature estimation adapted from [27]

Figure 8: Control volume used in Eq.3 and illustration of the error induced by using the average temperature to compute heat losses