An End-to-End Machine Learning Framework for District Heating Networks Simulation

Taha Boussaid, Vasile-Marian Scuturici, François Rousset, Marc Clausse

An End-to-End Machine Learning Framework for District Heating Networks Simulation

Taha Boussaid
taha.boussaid@insa-lyon.fr
Univ. Lyon, INSA Lyon,
CNRS, LIRIS, UMR 5205 & CETHIL, UMR 5008
Villeurbanne, FRANCE

Vasile-Marian Scuturici
Univ. Lyon, INSA Lyon,
CNRS, LIRIS, UMR 5205
Villeurbanne, FRANCE

François Rousset
Univ. Lyon, INSA Lyon,
CNRS, CETHIL, UMR 5008
Villeurbanne, FRANCE

Marc Clausse
Univ. Lyon, INSA Lyon,
CNRS, CETHIL, UMR 5008
Villeurbanne, FRANCE

ABSTRACT

Faced with environmental challenges, district heating networks have been identified as a viable solution to decarbonize the heating sector. However, they raise various challenges regarding the optimization of their control given their size and the operational constraints of the energy systems involved. As a result, the numerical simulation of these networks is computationally heavy, which hinders near-instantaneous optimal control. In this work, we present the first brick of an optimization framework for the control of district heat networks using a surrogate model based on geometric deep learning. More precisely we trained specific architectures of Graph Neural Networks to emulate a thermo-hydraulic simulator of district heating network. This statistical inference method allows us to drastically reduce simulation time by 1 to 4 orders of magnitude.

KEYWORDS

datasets, graph neural networks, surrogate modelling

ACM Reference Format:
Taha Boussaid, Vasile-Marian Scuturici, François Rousset, and Marc Clausse. 2023. An End-to-End Machine Learning Framework for District Heating Networks Simulation. In *Proceedings of Conférence sur la Gestion de Données – Principes, Technologies et Applications (39th BDA conference).* Montpellier, France, 6 pages.

1 INTRODUCTION

In Europe, the production of heat and cold represents half of the energy consumption and is mainly based on fossil fuels [6]. In this context, the 4th generation of district heating networks (DHN) have been identified as a viable option to decarbonize the heat production sector [9, 7] as they take advantage of various heat sources including renewable and recycled ones such as biomass, geothermal, solar thermal, and waste heat. However, the use of various heat sources, some of which are intermittent, adds a new complexity to controlling the system including the stochastic character of some variables such as resources availability, weather conditions and electricity price. Therefore, an intelligent control strategy is required in order to optimally operate these systems, such that the operational costs are reduced, the energy efficiency is maximised and the environmental impacts are limited.

A detailed review on control strategies for DHN can be found in [16]. This study shows that all these strategies make use of several simulations because of the iterative nature of control algorithms that need to understand and predict the behavior of the system and its response to various scenarios of control variables. However, physical simulators of such systems are computationally heavy as they often need to solve non linear equations (e.g. hydraulic equations). Thus, accurate and yet fast numerical models of the network's different components and their interactions are needed. Our proposition to overcome this limitation is the formulation of a numerically efficient and stable surrogate model of DHN simulation. Recently, different studies investigated the application of machine learning (ML) algorithms as surrogate models for different energy systems [15, 10, 11]. For example, the study in [10] consisted on training a graph neural network (GNN) 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 classic optimization methods and exhibited much better scalability to larger networks.

In the field of DHN control optimization, ML was applied at two different levels. The first one focuses on predicting the thermal load [20, 18] and the second, more recent, applying deep reinforcement learning to train autonomous agents to optimally operate DHN [21]. However, to the best author's knowledge, no attempt to formulate a surrogate model of DHN simulation using ML have been made. In this paper, we present a complete pipeline that was developed for simulating DHN using spatio-temporal graph convolution neural networks (STGCN).

2 PROBLEM STATEMENT

The topology of a DHN can be defined using graph theory. In the development of DHN project, there are two major challenges. First a design problem, where the production power is sized with respect to the heat demand of all connected consumers, and the network topology is optimized with regards to heat losses and investment costs. The second, which is the subject of this work, is a control problem. In the latter, given a DHN with a specific topology, the objective is to simulate its dynamic response to different scenarios. A scenario corresponds to a set of exogenous and control variables. More precisely, we want to know the temperature and mass flow

39th BDA conference, October 23–26, 2023, Montpellier, France
2023.

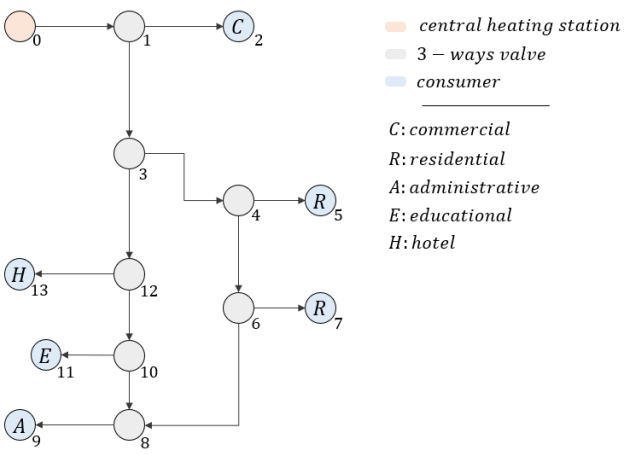

Figure 1: DHN topology of the study case adapted from [1]. The site includes buildings of different usage profiles.

rates at each node of the network. Throughout this study, the DHN shown in Figure 1 is used to evaluate our approach. This choice is based on the availability of data needed by the physical simulator. The site includes buildings of different usage profiles where the number under each node indicates its index.

Therefore, the topology of any DHN can be 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 3-ways valves (i.e. *T junctions*). More precisely, the network is completely defined through two variables:

- 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.
- Node type vector $V_0 \in \mathbb{R}^N$ that defines the type of each node with regard to a predefined set of supported node types within the physical simulator.

Once the topology is defined, to simulate the operation of the DHN, the physical simulator then requires other physical parameters to solve the thermo-hydraulic equations (Navier Stockes and Energy balance [4]). Mainly, the thermo-physical properties of each branch, the heat demand at each consumer node and the external temperature. The solver then outputs the evolution of the variables of interest through a given time horizon induced from the length of heat load matrix. In our case, the variables are: incoming and outgoing temperatures (T_f, T_r) and mass flow rates $\dot{m}$ at each node. To this purpose, and with a vision to publish it further as a data generator for DHN simulation, we developed a physical simulator. We used `TESPy` (Thermal Engineering Systems in Python) package [19], an open-source physical solver. To simulate both steady (i.e. constant regime) and transient (i.e. variable regime) dynamics of DHN, a customised code layer has been built upon this physical solver.

The inputs of the surrogate model are heterogeneous. They are distinguished according to their static or dynamic character and then according to whether they are local or global, in other terms whether they are defined at nodes/edges levels or at the graph level. First, let $\mathcal{H}$ be the horizon of prediction of the surrogate model (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 types to a predefined scalars set that don't change over time. 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, they remain constants during the simulations. Subsequently, this variable is noted $E_s \in \mathbb{R}^{N_b \times 3}$.
- Dynamic nodes attributes: it consists of a matrix containing the temporal evolution of heat demand at each node, it will be noted $X_d \in \mathbb{R}^{N \times p}$. For T-junctions 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 $T_{ext} \in \mathbb{R}^p$.

Finally, the inference problem can be stated as a supervised learning problem schematized in figure 2. The network topology is implicitly included during the learning phase via the adjacency list that is used by the graph attention mechanism (explained in section 3.1) in order to select the neighbors of each node.

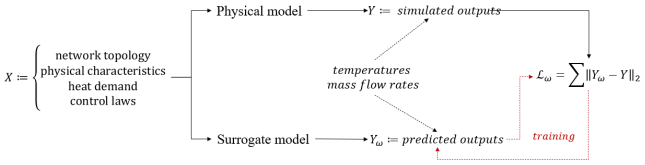

Figure 2: Supervised learning problem to train the surrogate model

3 SURROGATE MODELS

During the development phase of this framework, different types of models were evaluated for the studied problem. Here we present the two architectures offering the best performance. The choice of these architectures is the result of an extensive literature study on surrogate models and in particular graph neural networks [2, 13, 3, 8, 22, 17]. The following sections show the evaluated architectures. They were implemented using `Pytorch geometric` package [5] and its extension for temporal models [12].

3.1 Encoder-Decoder model

The first architecture that has been explored is in the form of an encoder-processor-decoder as shown in figure 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 [17]. In the graph attention layer, the hidden representation of each node i is computed as a weighted sum of its neighbors' $j \in \mathcal{N}(i)$ representations, where the weights are learned using an attention mechanism (Eq 1). Finally the decoder is composed of two linear layers that maps the nodes representation to the output space. operation is written as:

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

where $\alpha_{i,j}$ is the attention coefficient for each neighbor j of node i and σ a non-linear activation function.

3.2 Recurrent model

The inference problem involves data with temporal dependencies. It is known that recurrent neural networks are suited to this type of data. Therefore, the second architecture that was implemented is a spatio-temporal graph convolution neural network (STGCN) inspired from [22]. In addition to the spatial convolution where the nodes exchange information with their neighbors, this model incorporates a temporal cell based on Gated Recurrent Unit (GRU) in order to update the node representations using their previous hidden states too. In figure 4, *GC* refers to graph convolution, which updates the node features spatially before feeding them to the GRU cell that consists of an update u_t gate, a reset gate r_t and a candidate new hidden state c_t.

$$\tilde{x}_t = \text{GraphConv}\,(x_t) \tag{2}$$

$$r_t = \sigma\,(W_r \cdot [\tilde{x}_t,\ h_{t-1}] + b_r) \tag{3}$$

$$c_t = \tanh\,(W_c \cdot [\tilde{x}_t,\ r_t \cdot h_{t-1}] + b_c) \tag{4}$$

$$u_t = \sigma\,(W_u \cdot [\tilde{x}_t,\ h_{t-1}] + b_u) \tag{5}$$

$$h_t = u_t \cdot h_t + (1 - u_t) \cdot c_t \tag{6}$$

4 RESULTS AND DISCUSSION

The two architectures above are evaluated agains the physical model on the study case of the Figure 1. The site includes buildings of different usage profiles. Given the annual heat demand for each consumer and a typical external temperature variation, an hourly consumption profile is generated for each building using the geometric series approach implemented in `demandlib` library [14], a Python package that allows to create heat profiles from annual values. The control variables are chosen to be the supply temperature and the total mass flow rate leaving the producer node.

The network operation is then simulated for one year with a time step $\Delta t = 10min$. The outputs are time series of $8760 \times 6 = 52560$ data points for each node, representing the evolution of temperatures. The simulation time using the physical model is equal to $t_{sim,4c}$ = 3h2min using a 4-cores CPU processor with a 32-GB RAM. For the encoder-decoder model, we decided a split of 75% for training and 25% for validation. As this model needs a predefined time horizon of prediction, it is set to 1 week = $7 \times 24 \times 6 = 1008$ data points. Thus, 39 weeks are used for training and 13 weeks for validation. On the other hand, as the STGCN is an auto-regressive model, we do not need to define the prediction horizon, and the same splitting is used. Without loss of generality, here we show the predictions for return temperatures only. Moreover, we plot its normalized value:

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

First, the models were evaluated on data with slow frequency control laws, i.e. supply temperature and the total mass flow rate remain almost constant for a given time span. In this case the variable having the strongest weight on the values to be predicted (forward and return temperatures) is the heat demand profile. In this case, both models performed very well in predicting the variables of interest. An example is given in Figure 5. 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 0.7 Kelvin (0.7K) for all nodes, which is acceptable for the end users, i.e. network operators.

In order to test further the models adaptability and sensitivity to control laws, we tested them 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 Encoder-Decoder 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. On the other hand, the STGCN model captured well enough the system dynamics as shown in Fig. 7. This is the direct result of using GRU to better incorporate the notion of temporality and time dependence into the inference function.

In Figure 8 we compare the two models using RMSE over each node in the validation dataset. This confirms that temporal dependencies are a key feature to have in the surrogate model as the STGCN clearly outperforms the Encoder-Decoder model. This result is consistent with the physics of DHN, which are known to have a big inertial effect. However, the inference time is different for both models. To simulate one week of operation, the Encoder-Decoder model needs $t_{inference} = 0.019s$ against $t_{sim,4c} = 336s$ for the physical simulation on the validation dataset which accounts for approximately 1.8×10^4 time gain. On the other hand, the STGCN model needs $t_{inference} = 7.71s$, this accounts for approximately 43 time gain. Finally, although these results need further analysis, and deeper sensitivity and ablation studies, they suggest that a combination of the two models with regards to the variations of the control laws may be a good compromise between fast computation and accurate predictions.

5 CONCLUSION

In this paper, we presented a work in progress for the application of geometric deep learning as framework for surrogate modeling of district heating networks simulation. The bottleneck to control such systems is their heavy simulations. Therefore, the aim of the surrogate model is to reduce computation time while preserving a high accuracy. Depending on the architecture, the time gain varies in our experiences from a factor of 43 to 1.8×10^4. The next line of work

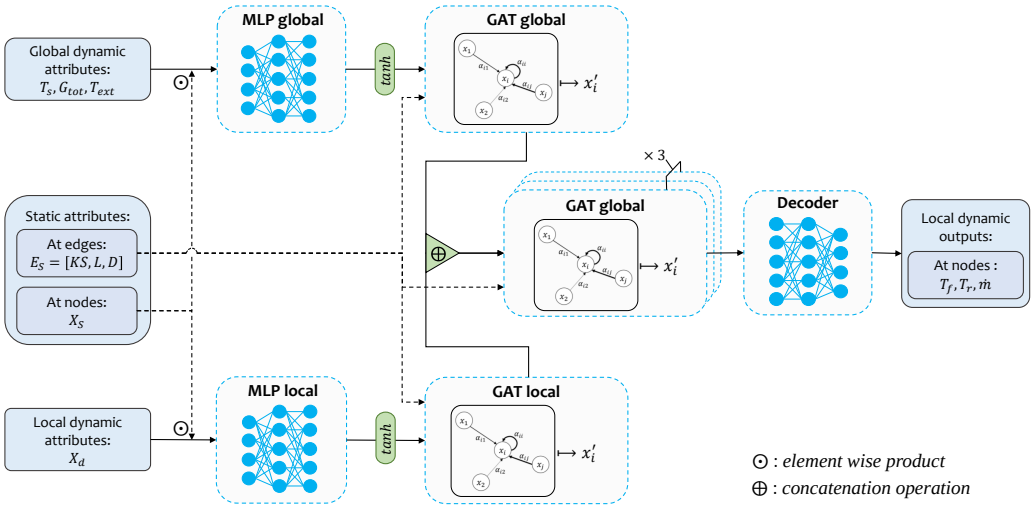

Figure 3: Surrogate model encoder-processor-decoder architecture

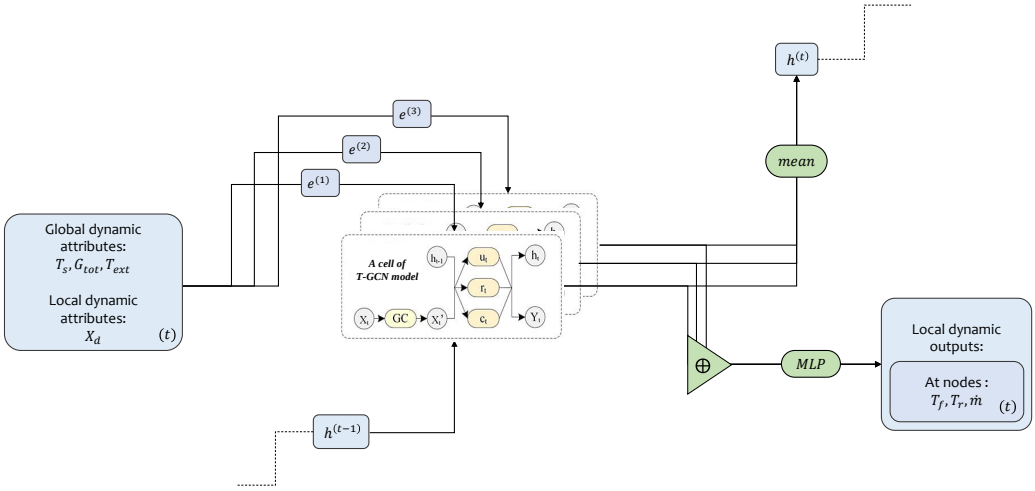

Figure 4: Spatio-temporal model architecture. The STGCN cell schematic is adapted from [22].

Table 1: RMSE for return temperatures prediction on validation dataset for the Encoder-Decoder model

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

is to deepen the analysis of these models and to fuse more physical constraints during the training phase to respect the physical laws in the predicted values.

ACKNOWLEDGMENTS

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

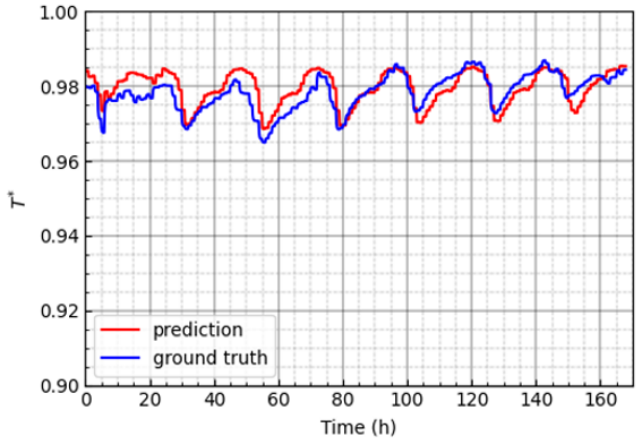

Figure 5: Normalized temperature (T^*) prediction over 1 week for node 13 using Encoder-Decoder model with moderate control laws, in this case the supply temperature remains constant.

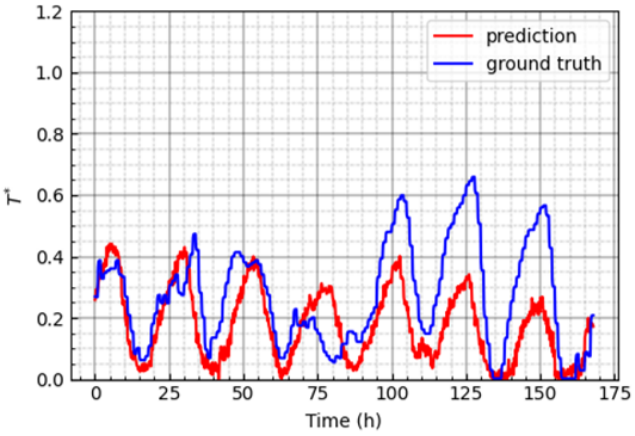

Figure 6: Normalized temperature (T^*) prediction over 1 week for node 13 using Encoder-Decoder model and with sharp control laws. The model doesn't capture well the dynamics of the DHN system.

REFERENCES

[1] Francesca Belfiore. 2021. District heating and cooling systems to integrate renewable energy in urban areas. Tech. rep. EPFL.

[2] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. 2021. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*.

[3] Ameya Daigavane, Balaraman Ravindran, and Gaurav Aggarwal. 2021. Understanding convolutions on graphs. *Distill*, 6, 9, e32.

[4] Itzal del Hoyo Arce, Saioa Herrero López, Susana López Perez, Miika Rämä, Krzysztof Klobut, and Jesus A Febres. 2018. Models for fast modelling of district heating and cooling networks. *Renewable and Sustainable Energy Reviews*, 82, 1863–1873.

[5] Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.

[6] T Fleiter et al. 2017. Heat roadmap europe: eu profile of heating and cooling demand in 2015. *Heat roadmap EU*.

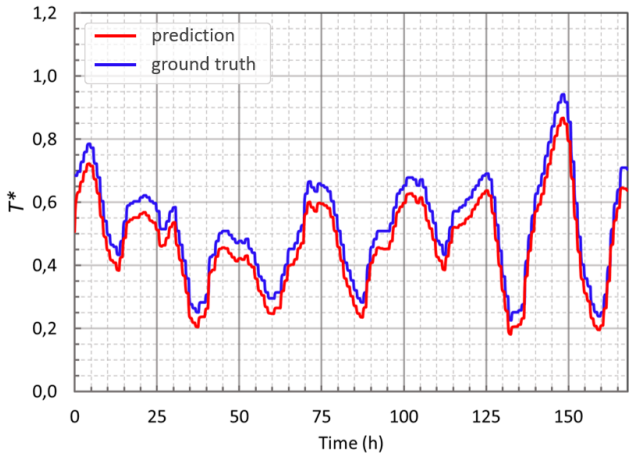

Figure 7: Normalized temperature (T^*) prediction over 1 week for node 13 using STGCN model and with sharp control laws

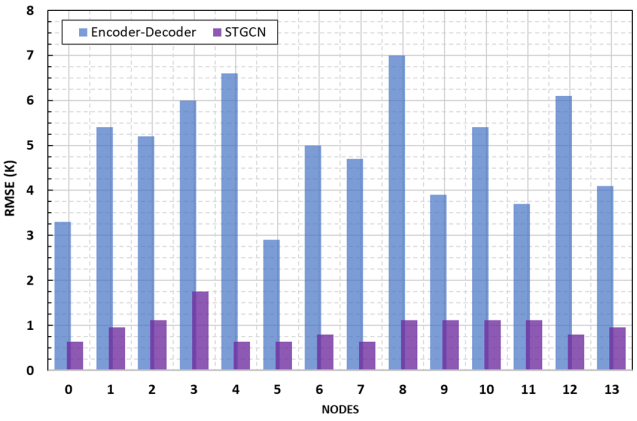

Figure 8: Comparison of Encoder-Decoder and STGCN performances for each node in the studied DHN.

[7] Haoran Li and Natasa Nord. 2018. Transition to the 4th generation district heating-possibilities, bottlenecks, and challenges. *Energy Procedia*, 149, 483–498.

[8] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

[9] Henrik Lund, Sven Werner, Robin Wiltshire, Svend Svendsen, Jan Eric Thorsen, Frede Hvelplund, and Brian Vad Mathiesen. 2014. 4th generation district heating (4gdh): integrating smart thermal grids into future sustainable energy systems. *Energy*, 68, 1–11.

[10] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. 2020. Optimal power flow using graph neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5930–5934.

[11] Yi Ming Ren, Mohammed S Alhajeri, Junwei Luo, Scarlett Chen, Fahim Abdullah, Zhe Wu, and Panagiotis D Christofides. 2022. A tutorial review of neural network modeling approaches for model predictive control. *Computers & Chemical Engineering*, 107956.

[12] Benedek Rozemberczki et al. 2021. PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 4564–4573.

[13] Benjamin Sanchez-Lengeling, Emily Reif, Adam Pearce, and Alexander B Wiltschko. 2021. A gentle introduction to graph neural networks. *Distill*, 6, 9, e33.

[14] Möller C et al Schachler B. 2021. Demandlib: creating heat and power demand profiles from annual values. https://oemof.org/libraries/#demandlib.

[15] Lei Sun, Tianyuan Liu, Ding Wang, Chengming Huang, and Yonghui Xie. 2022. Deep learning method based on graph neural network for performance prediction of supercritical co2 power systems. *Applied Energy*, 324, 119739.

[16] Annelies Vandermeulen, Bram van der Heijde, and Lieve Helsen. 2018. Controlling district heating and cooling networks to unlock flexibility: a review. *Energy*, 151, 103–115.

[17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Pietro Romero Adriana and Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[18] Paul Westermann, Matthias Welzel, and Ralph Evins. 2020. Using a deep temporal convolutional network as a building energy surrogate model that spans multiple climate zones. *Applied Energy*, 278, 115563.

[19] Francesco Witte and Ilja Tuschy. 2020. TESPy: Thermal Engineering Systems in Python. *Journal of Open Source Software*, 5, 49, 2178. DOI: 10.21105/joss.02178.

[20] Jiyang Xie, Hailong Li, Zhanyu Ma, Qie Sun, Fredrik Wallin, Zhongwei Si, and Jun Guo. 2017. Analysis of key factors in heat demand prediction with neural networks. *Energy Procedia*, 105, 2965–2970.

[21] Bin Zhang, Amer MYM Ghias, and Zhe Chen. 2022. 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*, 8, 15067–15080.

[22] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: a temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21, 9, 3848–3858.