




MILENA PETKOVIC¹, THORSTEN KOCH², JANINA ZITTEL³

Deep learning for spatio-temporal supply and demand forecasting in natural gas transmission networks

¹  0000-0003-1632-4846
²  0000-0002-1967-0077
³  0000-0002-0731-0314

Zuse Institute Berlin
Takustr. 7
14195 Berlin
Germany

Telephone: +49 30 84185-0
Telefax: +49 30 84185-125

E-mail: bibliothek@zib.de
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Deep learning for spatio-temporal supply and demand forecasting in natural gas transmission networks

Milena Petkovic ², Thorsten Koch ^{1, 2}, and Janina Zittel ²

¹Technische Universität Berlin, Chair of Software and Algorithms for Discrete Optimization,
Straße des 17. Juni 135, 10623 Berlin, Germany

²Zuse Institute Berlin, Departement for Applied Algorithmic Intelligence Methods,
Takustraße 7, 14195 Berlin, Germany

December 2020

Abstract

Germany is the largest market for natural gas in the European Union, with an annual consumption of approx. 95 billion cubic meters. Germany's high-pressure gas pipeline network is roughly 40,000 km long, which enables highly fluctuating quantities of gas to be transported safely over long distances. Considering that similar amounts of gas are also transshipped through Germany to other EU states, it is clear that Germany's gas transport system is essential to the European energy supply. Since the average velocity of gas in a pipeline is only 25km/h, an adequate high-precision, high-frequency forecasting of supply and demand is crucial for efficient control and operation of such a transmission network. We propose a deep learning model based on spatio-temporal convolutional neural networks (DLST) to tackle the problem of gas flow forecasting in a complex high-pressure transmission network. Experiments show that our model effectively captures comprehensive spatio-temporal correlations through modeling gas networks and consistently outperforms state-of-the-art benchmarks on real-world data sets by at least 21%. The results demonstrate that the proposed model can deal with complex nonlinear gas network flow forecasting with high accuracy and effectiveness.

1 Introduction

Germany is the largest market for natural gas in the European Union, with an annual consumption of approx. 95 billion cubic meters [1]. Since a similar amount of gas is also transported across Germany to other EU states, Germany represents a vital transit hub for natural gas [11]. Natural gas is the second most used energy source in Germany, and in 2019, its share in Germany's primary

energy consumption was 25 % [34]. Even though the heat market is still by far the most important market for natural gas, recently, natural gas is being used for other purposes as well. In particular, gas plays an important role in the German “Energiewende”, especially in the transition from fossil fuels to renewables in the power sector. Due to short ramp-up times, gas-power-plants provide a fast source of electricity in times of low availability of renewable sources.

The German high-pressure gas pipeline network’s length is roughly 40,000 km, enabling highly fluctuating quantities of gas to be transported efficiently and safely over long distances. Considering that large amounts of gas are also transshipped to other countries, it is clear that the gas transport system is essential to the European energy supply. Expansion measures are cost-intensive and take long time to implement. Thus, the optimal use of existing capacities is crucial for safety of supply and service quality. Since the average gas pipe velocity is only 25km/h, adequate high-precision and high-frequency forecasting of supply and demand is a decisive matter for efficient control and operation of the complex natural gas transmission networks and distribution systems.

Traditionally, statistical models have been commonly used as a tool for natural gas forecasting. In their early work Herbert et al. [23, 22] used regression analysis, residual analysis and linear regression equation to analyse effect of heating degree days (HDD), price of natural gas, price of residual fuel oil, and industrial activity on industrial demand for natural gas. Aras and Aras [3] proposed two individual autoregressive time series models for heating and non-heating season instead of attempting to capture the seasonal patterns in a single model while Gorucu [18] proposed multivariable regression for forecasting gas consumption. Timmer and Lamb [40] used linear model external variables like days below percentile (DBP) and heating degree days (HDD) in determining relations between temperature and residential natural gas consumption. Vondracek et al. (2008) proposed statistical approach to estimate natural gas consumption of individual residential and small commercial customers using nonlinear regression. Autoregressive models are also commonly used in energy forecasting. ARIMA models were reported in Liu and Lin [32] and Erdogan [14] while Brabec [8] used Nonlinear mixed effects model with individual customer-specific parameters and compared the model with ARIMAX (Autoregressive integrated moving average with eXtra / eXternal process) and ARX (Auto-regressive with exogenous inputs). Recently, Chen et al. [12] proposed advanced Functional AutoRegressive model with eXogenous variables (FARX) for day-ahead hourly gas flow forecast.

Especially in the last two decades machine learning methods have intensively been developed to model more complex data. Kizilaslan and Karlik [28, 29] used ANN with several different algorithms (Quick propagation Algorithm, Conjugate Gradient Descent Algorithm, Levenberg-Marquardt Algorithm etc.) to forecast daily, weekly and monthly gas consumption in Istanbul. They used regression analysis to determine effect of different factors. Szpilk [39] presented Multilayer Perceptron(MLP) with additional calendar and weather features for predicting natural gas consumption in the city Szczecin in Poland. Nabavi et al. [33] developed and compared multiple linear regression, logarithmic multiple

linear regression methods, and nonlinear autoregressive with exogenous input artificial neural networks (NARAX) model as a tool for predicting annually consumption in residential and commercial sectors in Iran. Many research studies showed that hybrid models (combining or ensembling) improve performance and robustness of forecast relative to individual forecast models. This is especially important when forecasting multiple time series with different behaviour (such as nodes in the gas network representing different demands of suppliers, industrial or residential customers) since hybrid models enhance the advantages of the individual approaches [35, 42, 25].

It is important to notice that, although the traditional statistical and machine learning models generally have good forecast accuracy in real-world data applications, they often neglect the information available in the spatial-temporal correlations of high-dimensional, complex gas network data. Moreover, the prediction performances of traditional methods depend profoundly on feature engineering, which usually requires expert experience in the corresponding field.

In recent years, deep learning methods have been employed to deal with high-dimensional spatio-temporal data defined on the complex non-Euclidean domains such as graphs. The success of convolutional neural networks (CNN) in extracting spatial features of grid-based data inspired advances in redefining the notation of convolution and adapt to the graph domain. Bruna et al. [10] were pioneers in developing a graph convolution based on the spectral graph theory. Following their work, several researchers improved and extended spectral-based convolutional graph neural networks [31, 13, 27]. In their survey Wu et al. [43] give a comprehensive overview of graph neural networks in data mining and machine learning fields together with an overview of possible applications of graph neural networks to various domains such as computer vision [44, 24], natural language processing [38, 5], traffic [47, 19, 45], chemistry [46], etc...

Inspired by the success of recently proposed deep learning models for energy time series forecasting [26, 36] we propose a deep learning spatio-temporal predictive model (DLST) to tackle the problem of gas flow forecasting in a complex high-pressure transmission network. In our model, the convolution is employed directly on graph-structured data to extract profoundly meaningful patterns and features in the space and time domain. We formulate the problem of forecasting gas flows on graphs and build the complete convolutional model, enabling us to significantly speed up training while using fewer parameters.

Our experiments show that our model effectively captures comprehensive spatio-temporal correlations through modeling gas networks. It consistently outperforms state-of-the-art benchmarks on real-world data sets for four different forecasting horizons with a skill score (calculated on normalized mean absolute percentage error) between 0.218 and 0.494. These results strongly indicate that the proposed model is capable of dealing with complex nonlinear gas network flow forecasting with high accuracy and effectiveness.

This work is part of a joint project with one of the Germany’s biggest transmission system operators (TSO), Open Grid Europe [17]. Together with OGE, we develop a tool for high precision gas flow forecasting, providing multi-step ahead hourly forecasts for more than 100 nodes in the gas network. These nodes

have very different statistical characteristics, as they represent a variety of different source and consumption points, ranging from connections to other gas networks or countries over industrial consumers to storage facilities.

The paper is organized as follows: In Section 1 we give a brief introduction to the motivation of the paper and presents related work. Section 2 gives a formulation of the problem and explains the methodologies used in building the DLST model. In Section 3 we introduce the real-world data set we use for the computational verification of our model described in Section 4. In Section 5 we discuss the obtained results. Finally, Section 6 brings some conclusions.

2 Forecasting on graphs

We formulate the problem of forecasting gas flow in the gas transmission network as follows: predict gas flow $(\hat{q}_{t+1}, \dots, \hat{q}_{t+H})$ in the next H time steps, where $q_t \in \mathbb{R}^n$ is a vector of gas flow values of n nodes at time step t , given the set of features $F \in \mathbb{R}^{d \times n}$ consisting of M historical flow observations (q_{t-M+1}, \dots, q_t) and an (optional) set of exogenous features $(ex_1, \dots, ex_{d-M}) \in \mathbb{R}^{(d-M) \times n}$ such as weather information, season indicators, economical parameters etc.

The observation q_t can be considered as a signal defined on the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{A}, W)$ which consists of a finite set of nodes \mathcal{V} with $|\mathcal{V}| = n$ representing the nodes of a gas transmission network, a set of arcs \mathcal{A} and an adjacency matrix $W \in \mathbb{R}^{n \times n}$ modeling the connections between nodes. If arc $a_{i,j}$ connecting vertices i and j exists, $W_{i,j}$ represents the weight of the arc; otherwise, $W_{i,j} = 0$.

For modeling connections between nodes, we split the set of arcs \mathcal{A} into individual sets $\mathcal{A} = \mathcal{A}^p \cup \mathcal{A}^a$ for the different types of network elements considered in this paper i.e. passive and active elements respectively.

The basic passive elements of the transmission network connecting two nodes are pipes. Nodes can be also connected via active elements such as valves, regulators and compressor stations that can potentially change the network topology and are used for controlling pressure along the direction of the flow, see [21].

Furthermore, we split the set of nodes $\mathcal{V} = \mathcal{V}^b \cup \mathcal{V}^o$ into a set of boundary nodes and a set of inner nodes, respectively. We denote $\mathcal{V}^+ \subseteq \mathcal{V}$ the *supply* and $\mathcal{V}^- \subseteq \mathcal{V}$ the *demand* nodes of the network.

Using the information of graph data, we can capture the spatio-temporal interdependence among nodes. The main challenge is that due to non-Euclidean nature of graph data some essential operations of machine learning such as convolutions do not apply to a graph domain [9]. Furthermore, for graph data, instances are not independent from each other since each instance (node) is related to others by its connections.

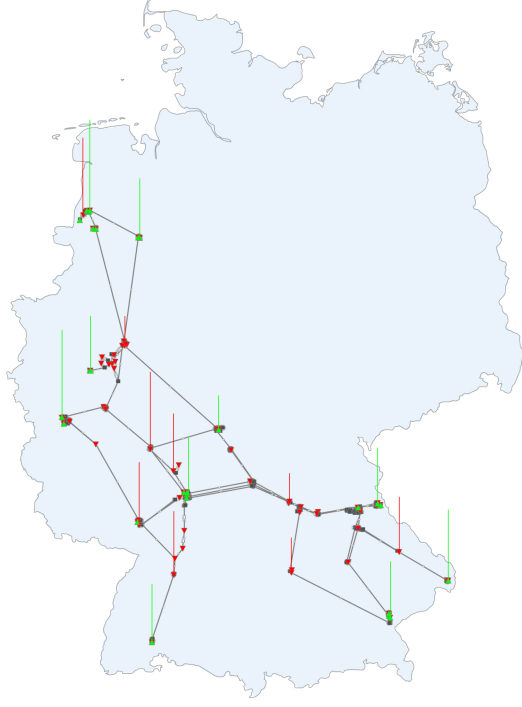


Figure 1: Illustration of graph-structured flow data. Each q_t represents a frame of current flow status at time step t

2.1 Spatio Graph Convolution Unit (SGCU)

First, let us define the spatial convolution on a given graph \mathcal{G} . We define the diagonal degree matrix as $D \in \mathbb{R}^{n \times n}$ with $D_{ii} = \sum_j W_{ij}$ and Laplacian as $L = I - D^{-\frac{1}{2}}WD$ where I is an identity matrix. Then the Singular Value Decomposition (SVD) is applied to Laplacian as $L = U\Lambda U^T \in \mathbb{R}^{n \times n}$ where matrix $U \in \mathbb{R}^{n \times n}$ consists of eigenvectors of normalized Graph Laplacian transforms a signal to frequency domain and $\Lambda \in \mathbb{R}^{n \times n}$ is a diagonal matrix of eigenvalues of L . Now we define the graph Fourier transform of the input graph signal $x \in \mathbb{R}^n$ as $Ux \in \mathbb{R}^n$ and the graph Fourier transform of a filter $g \in \mathbb{R}^n$ as $Ug \in \mathbb{R}^n$. Finally, we define the convolution of a graph signal x with filter g as

$$g *_{\zeta} x = U^T(Ug \odot Ux) \quad (1)$$

where we denote $*_{\zeta}$ as graph convolution operator and \odot represents an element wise product. By defining $w = Ug$ as the filter in frequency domain, the convolution can be rewritten as

$$g * \varsigma x = U^T(\text{diag}(w)Ux) \quad (2)$$

The above graph convolution requires a filter w to have the same size as the input signal x , which would be inefficient and hard to calculate when the graph has a large size ($\mathcal{O}(n^2)$ multiplications with graph Fourier basis).

To localize the filter and reduce the number of parameters, we restricted the kernel $\text{diag}(w)$ to a polynomial of Λ as $\text{diag}(w) = \Theta(\Lambda) = \sum_{k=0}^{K-1} \Theta_k \Lambda^k$, where $\Theta \in \mathbb{R}^K$ is a vector of polynomial coefficients [13].

K represents the kernel size of graph convolution and determines the maximum radius of the convolution from central nodes. The number of trainable parameters are restricted to K .

In graph signal processing Chebyshev polynomial $T_k(x)$ is traditionally used to approximate kernels for graph signal as a truncated expansion of order $K-1$ as

$$\Theta(\Lambda) \approx \sum_{k=0}^{K-1} \Theta_k T_k(\tilde{\Lambda}) \quad (3)$$

with $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I_n$ where λ_{max} denotes the largest eigenvalue of L [20]. Now, the graph convolution can then be rewritten as

$$\Theta * \varsigma x = \Theta(L)x \approx \sum_{k=0}^{K-1} \Theta_k T_k(\tilde{L})x \quad (4)$$

where $T_k(\tilde{L}) \in \mathbb{R}^{n \times n}$ is the Chebyshev polynomial of order k evaluated at the scaled Laplacian $\tilde{L} = 2L/\lambda_{max} - I_n$. Using the polynomial approximation and recursively computing K -localized convolutions we reduce the cost of (1) to $\mathcal{O}(K|\mathcal{E}|)$ as (4) shows [13].

Further, we extend the graph convolutions operator " ς " defined on graph signal $x \in \mathbb{R}^n$ to multi-dimensional tensors. First, for a signal with C_i channels $X \in \mathbb{R}^{n \times C_i}$, the graph convolution can be generalized as:

$$y_j = \sum_{i=1}^{C_i} \Theta_{i,j}(L)x_i \in \mathbb{R}^n, 1 \leq j \leq C_o \quad (5)$$

with $C_i \times C_o$ vectors of Chebyshev coefficients $\Theta_{i,j} \in \mathbb{R}^K$ (C_i, C_o are the size of input and output of the feature maps, respectively). The graph convolution for 2-D variables is denoted as " $\Theta * \varsigma x$ " with $\Theta \in \mathbb{R}^{K \times C_i \times C_o}$.

If the graph signal $X \in \mathbb{R}^{d \times n \times C_i}$, is a 3-D variable composed of d matrices whose column i is the C_i -dimensional value of X at the i^{th} node in graph \mathcal{G} we can apply the equal graph convolution operation with the same kernel Θ for each feature of d . Thus, the graph convolution can be further generalized in 3-D variables, noted as $\Theta * \varsigma \mathcal{X}$ with $\mathcal{X} \in \mathbb{R}^{d \times n \times C_i}$.

2.2 Temporal Convolution Unit (TCU)

Although recurrent neural networks (RNN) are still the most common choice for time series forecasting, CNN models recently achieved great success in dealing with time series data [16, 6] since they overcome main problems of recurrent models. The weights used to compute the output neurons in a feature map are the same (the same filter is used for each location) so the training of CNN is faster and with fewer parameters. In the case of a long input sequence, RNN use a significant amount of memory to store the partial results for their multiple cell gates while in CNNs the filters are shared across a layer, with the back-propagation path depending only on the network depth. Additionally, the number of parameter to learn is significantly reduced since each hidden neuron is connected only to a subset of input neurons that are close to each other.

To capture the temporal dynamic behavior of gas flows we employ convolutional neural network with a specific design suitable for time series data: Temporal convolutional networks (TCNs) inspired by [41, 4, 7] and recently wildly used in numerous applications [37, 30].

If we consider a 1-D time-series $x \in \mathbb{R}^n$ and a one-dimensional filter $\Phi \in \mathbb{R}^{K_t}$, we can define the i^{th} element of the convolution between x and Φ as:

$$f(i) = (\Phi * x)(i) = \sum_{j=0}^{K_t-1} x(i-j)w(j) \quad (6)$$

Instead of the standard convolution operator (Figure 2(a)), temporal convolution networks use causal convolutions which ensure that the prediction made at the time step t , \hat{q}_{t+1} can not depend on any of the future time steps feature f_{t+1} as shown in Figure 2(b).

Futhermore, TCNs use dilated convolutions defined as:

$$f(i) = (\Phi *_{dil} x)(i) = \sum_{j=0}^{K_t-1} x(i-dj)w(j) \quad (7)$$

where d is a dilation factor and $*_{dil}$ represents modified convolutional operator which apply the same filter over an area larger than its length by skipping input values with a certain step as it is shown in Figure 2(c). Note that with dilation $d = 1$ we have standard 1-d convolution. Using larger dilation enables an output at the top level to represent a wider range of inputs and effectively expanding the receptive field of a TCN.

The causal dilated convolution can be generalized to 3D variables by employing the same convolution kernel $\Phi \in \mathbb{R}^{K_t \times C_i \times C_o}$ to every node signal with C_i channels $x_i \in \mathbb{R}^{d \times C_i}$ in graph \mathcal{G} equally, noted as " $\Phi *_{dil} x$ " with $x \in \mathbb{R}^{d \times n \times C_i}$.

We form a Temporal Convolutional Unit (TCU) (as shown on Figure 3 in a pink block) by stacking causal dilated convolution layers with dilation factors in an increasing order which enables the receptive field of a model to grow exponentially. This way, TCU is capable to capture longer sequences with less

layers and saves computational resources. The TCU consist of L 1-D causal dilated convolution with a kernel K_t and dilation factors $d = 2^l, l \in [0, L - 1]$, followed by a non-linearity and a dropout layer for avoiding overfitting. Furthermore, residual connections are implemented among stacked temporal convolutional layers [41].

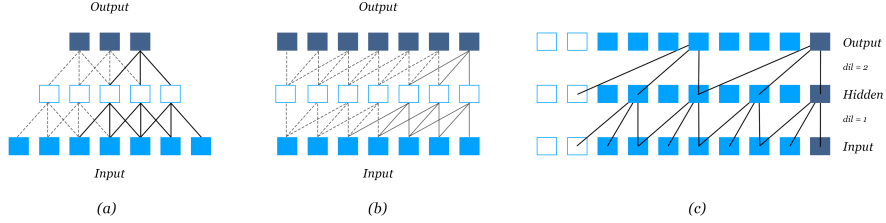


Figure 2: (a) Standard convolutional Network (b) Causal Convolutional Network with $K_t = 3$ (c) Causal Dilated Convolutional Network with $K_t = 3$ and $dil = 1, 2$

2.3 DLST architecture

In order to combine features from both spatial and temporal domains, we propose a DLST model to process the graph-structured gas flow time series.

As Figure 3 shows, in the first block of the model we process separately the historical flow information and the exogenous information through a standard temporal convolutional layer with 1D causal convolution, non-linearity unit (NLU) and a dropout layer. Later, the results are concatenated together and processed by the first Spatio Graph Convolution Unit (SGCU). Furthermore, we apply a Temporal Convolutional Unit, as described in Section 2.2, with L layers where each layer consists of a block with 1D dilated causal convolution with increasing dilation factor $d = 2^l, l \in [0, L - 1]$, a non-linearity unit (NLU) and a dropout layer to prevent overfitting. After TCU we apply once more SGCU in order to explore spatial dependencies on long term temporal information.

Finally, we apply once more temporal 1D- CNN with fully connected layers to calculate the final prediction.

3 Data

In this study we aim to forecast hourly natural gas inflows and outflows (supply and demand) in a high-pressure gas pipeline network operated by one of the largest transmission system operators in Germany, Open Grid Europe GmbH [17]. The observed gas network has 397 nodes in total, 25 supply (\mathcal{V}^+), 73 demand nodes (\mathcal{V}^-) as well as 299 inner nodes (\mathcal{V}^o). Even though we calculate the forecast for all nodes in the network, both boundary and inner nodes, when

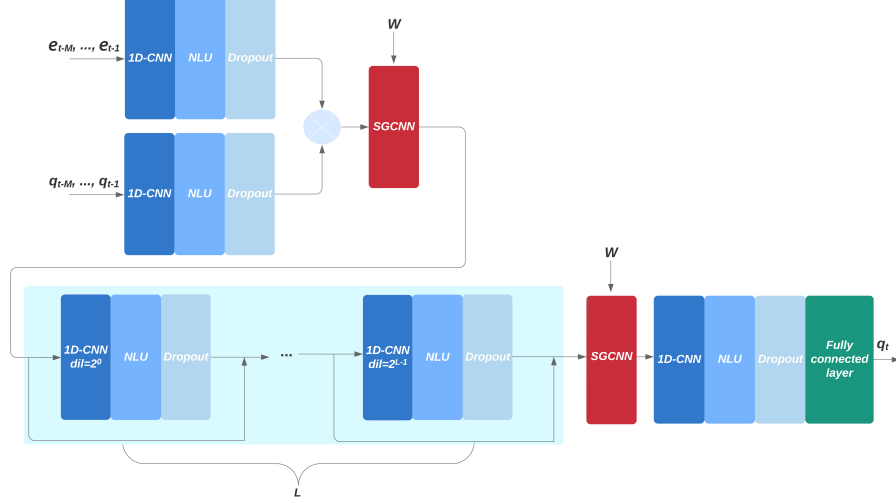


Figure 3: Architecture of DLST farmework

evaluating the forecast accuracy we focus only on the demand and supply nodes, since this is the main forecast goal.

For modeling the connections between the nodes we used the network topology provided by OGE. As described in Section 2, nodes can be connected with a passive (pipes) or an active element (valves, regulators, compressor stations). In the observed network the nodes are connected with 171 pipes, 254 valves, 154 regulators and 34 compressor stations.

We define the adjacency matrix W of the graph based on the distances among nodes in the gas network as follows:

$$w_{i,j} = \begin{cases} l_{i,j}, & \text{if } a_{i,j} \in \mathcal{A}^p \\ 0.01, & \text{otherwise} \end{cases} \quad (8)$$

where $l_{i,j} \in \mathbb{R}$ is the length of the pipe connecting nodes i and j .

The data set covers 33 months (May 2016- January 2019) of hourly gas flows for each node. Thus, every node of the gas network graph contains 24144 observations (covering 1006 days) in total. Additionally, we use as exogenous features, a set of measured air temperatures at the nodes location with the same time resolution. In order to compare results for different nodes, the data inputs are standardized by Z-Score method. Figure 4 shows summary statistics of 25 supply and 73 demand nodes. The nodes are sorted according to total absolute flow during the observed period and denoted as 'S' for *supply* nodes 'D' for *demand* nodes. The distribution of most of the nodes is centered around zero having wide-ranging variations. We can notice that some of nodes are skewed (both left and right) and have a large amount of outliers.

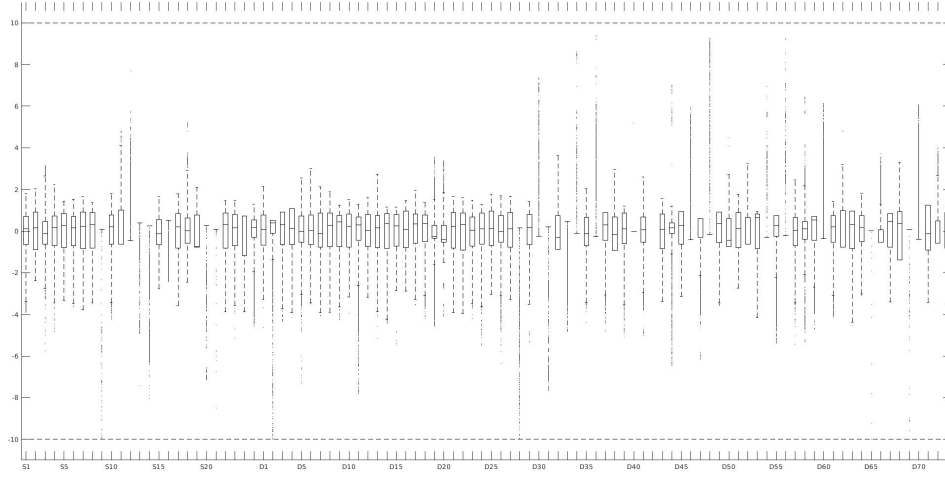


Figure 4: Boxplot of standardized gas flow of 25 supply and 73 demand nodes for period May 2016 to January 2019

Figure 5 represents the heatmap of standardized aggregated daily flow for the 98 boundary nodes. Large supply nodes (S1-S8) have a stable flow with very small sensitivity to season changing. These nodes usually represent transfer points from and to other networks and the flow is mainly induced by long term trading contracts. Other, smaller supply nodes show some seasonal patterns, especially high peaks during the winter period. Most of the demand nodes show strong seasonal dependency since they belong to the group of so called "Municipal supply nodes". These nodes supply residential areas and small commercial customers. Some of the largest (by total flow) demand nodes do not show a seasonal pattern, yet mostly intermittent behaviour and they belong to the group of Power Plant nodes, where a large amount of gas is needed in a relatively short period of time.

To transform the time series forecasting problem into a supervised machine learning problem we adopt a sliding window approach as illustrated in Figure 6. The window slides and obtains an input-output instance at each position. The output window size is set according to desired forecasting horizon and the optimal value for input window size had to be carefully decided based on the characteristics of the data sets and the model.

4 Forecasting procedure

We verify the performance of the proposed DLST model on the real-world data with hourly resolution between May 2016 and January 2019, as described in Section 3. The sample period is split as follows:

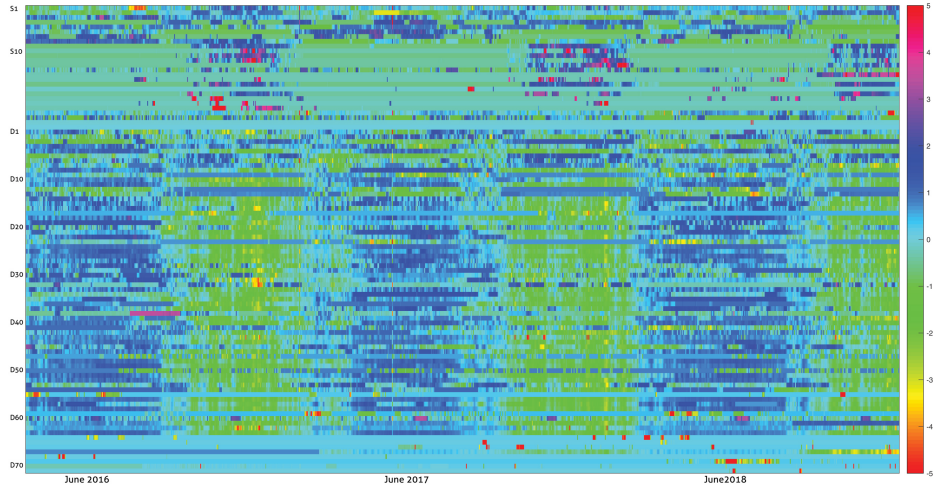


Figure 5: Heatmap of standardized daily flow of 25 supply and 73 demand nodes for period May 2016 to January 2019

Set	Name	From	To	Days
training	$T_{training}$	May 2016	September 2018	944
validation	$T_{validation}$	October 2016	November 2018	61
test	T_{test}	December 2018	January 2018	62

We train the model and optimize hyperparameters on the training and validation set by minimizing the square loss. We calculate out-of-sample forecasts on the test set using the fitted model and the rolling window method with the input of previous 24 observations and the output that is determined by the forecasting horizon. During the test, we do not update the hyperparameters of the model.

For each boundary (supply and demand) node we calculate an out-of-sample 1, 3, 6 and 12 hours ahead forecast for period 1.12.2018 until 31.01.2019 (62 days in total) with rolling windows approach.

Architecture and hyperparameters of DLST model are optimized using OPTUNA software [2]. Both the graph convolution kernel K_s and temporal convolution kernel K_t are set to 3 and for NLU we choose standard ReLU function. The number of channels are 32 for the input and first SGC unit, 64 for TCU as well as for second SGC unit. The TCU has 4 layers with dilation factor $dil = 2^l$, $l = 0, \dots, 3$. The proposed model is trained by minimizing mean squared error using ADAM optimizer for 50 epochs with a batch size of 32. The learning rate is initially set to 0.001 and was decayed for 0.9 every 5 epochs.

4.1 Benchmark models

We compare the performance DLST with several well-known benchmarks: ARIMA and the state-of-the-art artificial neural network models like MLP, CNN and

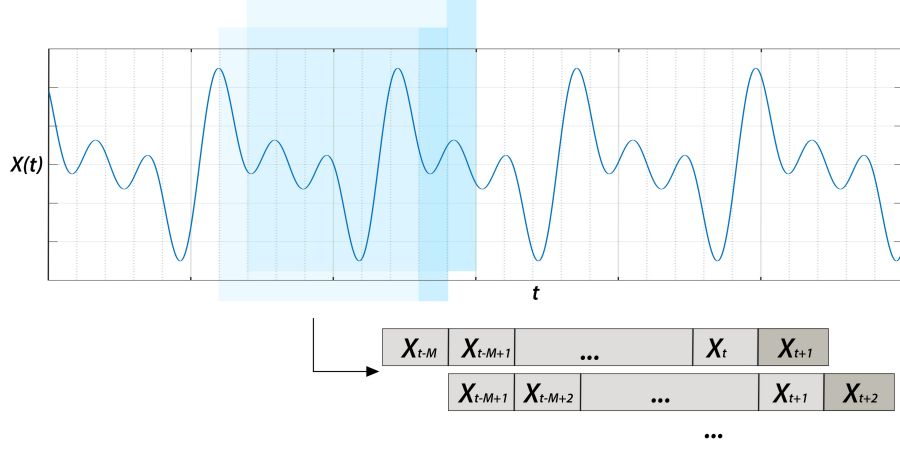


Figure 6: Sliding-window approach for data transformation.

LSTM [15] model. We determine the best ARIMA models for an univariate time series of 98 nodes according to a Akaike information criterion (AIC). To estimate the parameters of the ARIMA models we use input data of 28 days rolling window (parameter estimation of the ARIMA model is significantly faster then for neural network models). The LSTM is implemented using one LSTM layer hidden layer with optimized number of hidden neurons and learning rate for each individual node, a dropout of 0.1 followed by a fully connected output layer with the number of neurons equal to the forecast horizon and was trained for 100 epochs. MLP and CNN models are implemented in a similar fashion and optimized for the individual nodes. The input data for MLP, CNN and LSTM model is constructed in the same fashion as for DLST model, respecting the receptive filed of DLST, so that both models can use the same amount of historical information.

4.2 Performance measurement

The performance of DLST model is measured and quantified by calculating the forecast accuracy for individual (boundary) nodes, as well as the mean for the entire network. Specifically, we use mean absolute deviation (MAD), root mean squared error (RMSE), mean absolute percentage error(MAPE) as well as normalized mean absolute percentage error(nMAPE) defined as follows:

$$MAD = \frac{1}{24 * H * |T_{test}|} \sum_{d \in T_{test}} \sum_{t=1}^{24} \sum_{h=0}^{H-1} |q_{d,t+h}, -q_{d,\hat{t}+h}|$$

$$RMSE = \sqrt{\frac{1}{24 * H * |T_{test}|} \sum_{d \in T_{test}} \sum_{t=1}^{24} \sum_{h=0}^{H-1} (q_{d,t+h} - q_{d,\hat{t}+h})^2}$$

$$MAPE = \frac{1}{24 * H * |T_{test}|} \sum_{d \in T_{test}} \sum_{t=1}^{24} \sum_{h=0}^{H-1} \left| \frac{q_{d,t+h} - \hat{q}_t}{q_{d,t+h}} \right|$$

$$nMAPE = \frac{1}{24 * H * |T_{test}|} \sum_{d \in T_{test}} \sum_{t=1}^{24} \sum_{h=0}^{H-1} \left| \frac{q_{d,t+h} - q_{d,\hat{t}+h}}{\max(q)} \right|$$

where $q_{d,t,h}$ and $q_{d,\hat{t},h}$ are the real and forecast values of the natural gas flows on day d , hour t and H is a forecasting horizon. In order to compare the proposed method with alternative benchmark models we described in Section 4.1 we compute a different skill scores for each of the benchmark model :

$$skillscore(MAD)_{model_i} = 1 - \frac{MAD(DLST)}{MAD(model_i)}$$

$$skillscore(RMSE)_{model_i} = 1 - \frac{RMSE(DLST)}{RMSE(model_i)}$$

$$skillscore(MAPE)_{model_i} = 1 - \frac{MAPE(DLST)}{MAPE(model_i)}$$

$$skillscore(nMAPE)_{model_i} = 1 - \frac{nMAPE(DLST)}{nMAPE(model_i)}$$

A perfect forecast results in a skill score value of 1. A forecast which is outperforming the benchmark model will have a skill score between 0 and 1. If the skill is equal to the benchmark forecast the skill score will be 0, and a forecast which is under performing the benchmark forecast will have unbounded negative skill score values.

5 Results and discussion

In Table 1 we report the average skill scores of the DLST model compared to the four benchmark models for 25 supply and 73 demand nodes as well as the accuracy of the DLST model for reference. All metrics are calculated on the 1488 instances of the out-of-sample test set, as explained in Section 4. The proposed model outperforms all benchmark models for four considered forecasting horizons with a skill score of at least 0.218. Comparing to the "second best" model (marked bold in the Table 1), DLST has a skill score calculated with normalized mean absolute percentage error (nMAPE) between 0.218 and 0.494 depending on forecasting horizons. The proposed model shows high dominance in forecasting 3-steps ahead, while the difference is smallest for the one-step ahead horizon.

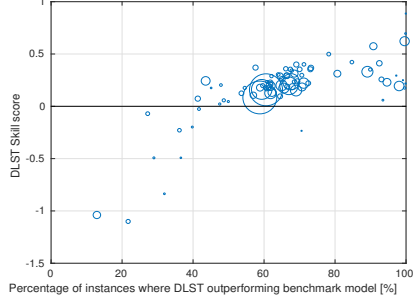
High mean accuracy (measured with four performance metrics) for every forecasting horizon and the positive skill score compared to all benchmark models strongly indicate that the DLST model more effectively captures comprehensive spatio-temporal correlations through modeling of the gas network. It

Model	DLST	ARIMA	MLP	CNN	LSTM
H	MAD	Skill score (MAD)			
1	0.012	0.450	0.565	0.658	0.264
3	0.018	0.460	0.532	0.575	0.435
6	0.028	0.423	0.426	0.443	0.296
12	0.043	0.384	0.350	0.297	0.308
	RMSE	Skill score (RMSE)			
1	0.028	0.768	0.326	0.492	0.155
3	0.040	0.749	0.327	0.395	0.282
6	0.055	0.723	0.242	0.282	0.198
12	0.076	0.697	0.175	0.148	0.174
	MAPE	Skill score (MAPE)			
1	0.032	0.517	0.433	0.630	0.405
3	0.048	0.502	0.471	0.506	0.399
6	0.076	0.455	0.324	0.392	0.280
12	0.122	0.410	0.234	0.236	0.193
	nMAPE	Skill score (nMAPE)			
1	0.012	0.489	0.516	0.616	0.218
3	0.018	0.543	0.517	0.530	0.494
6	0.027	0.512	0.411	0.402	0.336
12	0.041	0.447	0.307	0.231	0.264

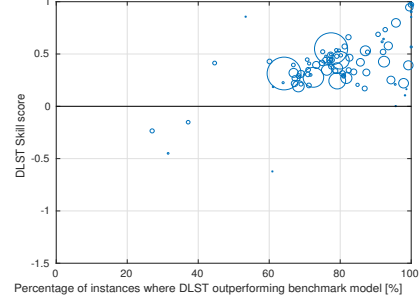
Table 1: Comparison of the DLST model and benchmark models. The first column shows different performance measures(MAD, RMSE, MAPE and nMAPE) achieved by DLST for forecasting horizons H=1,3,6 and 12. Other columns show skill scores for different benchmark models and performance measures.

benefits from the information available in the spatial-temporal correlations of high-dimensional, complex gas network data.

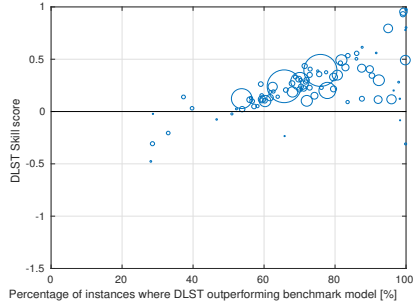
Figure 7 shows the average skill score for individual nodes and different forecasting horizons compared to the "second best" model. The circle size represents the size of the node based on the total amount of flow during the observed period, while the percentage of instances where DLST outperforms the benchmark is shown on the x-axes. While we can see that for some nodes proposed model has negative skill, it is clear that for most nodes, and especially nodes with biggest flows, the DLST model has a positive skill score and provides more accurate forecast for more than 50% of instances. The detailed ranking of the models for forecasting individual nodes is given in Figure 8. The ranking is calculated based on average nMAPE for individual nodes for every instance. DLST model outperforms the benchmark models for the first eight biggest supply and five biggest demand nodes in one step ahead forecasting. For longer horizons, especially H=3 and H=6, DLST is dominant, having the rank one for more then 85% of the nodes. LSTM is, in most of the cases, second most accurate model except for nodes like S15 and S18 where ARIMA model is having the rank one



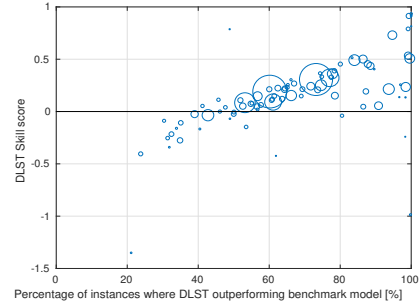
(a) H=1, Benchmark model LSTM



(b) H=3, Benchmark model LSTM



(c) H=6, Benchmark model MLP



(d) H=12, Benchmark model LSTM

Figure 7: Average DLST skill score (nMAPE) for individual nodes compared to "second best model". The percentage of instances where DLST model has a lower nMAPE than LSTM is shown on x-axes. The size of the circle denotes the total amount of absolute flow calculated for the period May 2016 - January 2019 for this node

for all horizons. The reason why neural network (machine learning) models are failing to forecast these two nodes lies in the fact that they had zero flows for more than 95% of the hours in the training set. As an illustration, Figure 9 shows measured and forecasted flow for 3 most significant supply and demand nodes (ranked by total absolute flow) for a testing period of two months.

6 Conclusion

In this paper, we have proposed a deep learning spatio-temporal model based on temporal and graph convolutional neural networks to capture both temporal and spatial dependencies in gas flows in a complex transmission network. The proposed model has been applied to predict hourly natural gas flows with different forecasting horizons (1,3,6 and 12) in one of Germany's longest and most

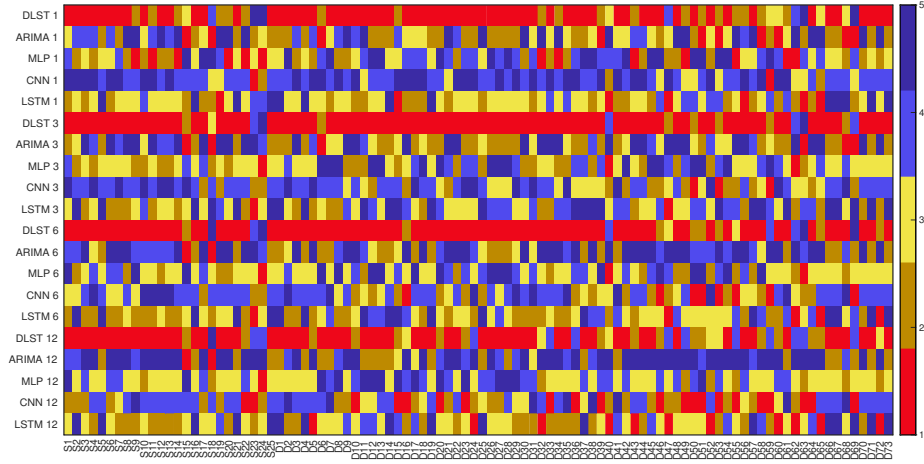


Figure 8: DLST and benchmark models (ARIMA, MLP, CNN and LSTM) ranked based on average nMAPE for individual nodes for different forecast horizons($H=1,3,6,12$)

complex pipeline network. We consider a network with gas flow time series of 98 boundary nodes, 25 supply and 73 demand nodes, and model the connections between the nodes to extract valuable information about mutual influences of different nodes in order to increase forecast accuracy. We have compared the proposed DLST model with several benchmark models popular in the literature. The obtained results show that the DLST model provides a stable and accurate forecast, consistently outperforming all considered benchmark models with a relative skill score between 0.218 and 0.494, especially in forecasting the most important nodes of the network. Among the benchmark models, the LSTM model tuned for individual nodes is the best one and gives a stable forecast accuracy, particularly for the nodes with strong seasonal patterns. Furthermore, there are special cases where the flow during the training period has significantly different behavior (e.g. high percentage of zero flow hours). For these cases, statistical models like ARIMA have an advantage compared to machine learning models.

Acknowledgement

The work for this article has been conducted within the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund numbers 05M14ZAM,05M20ZBM).

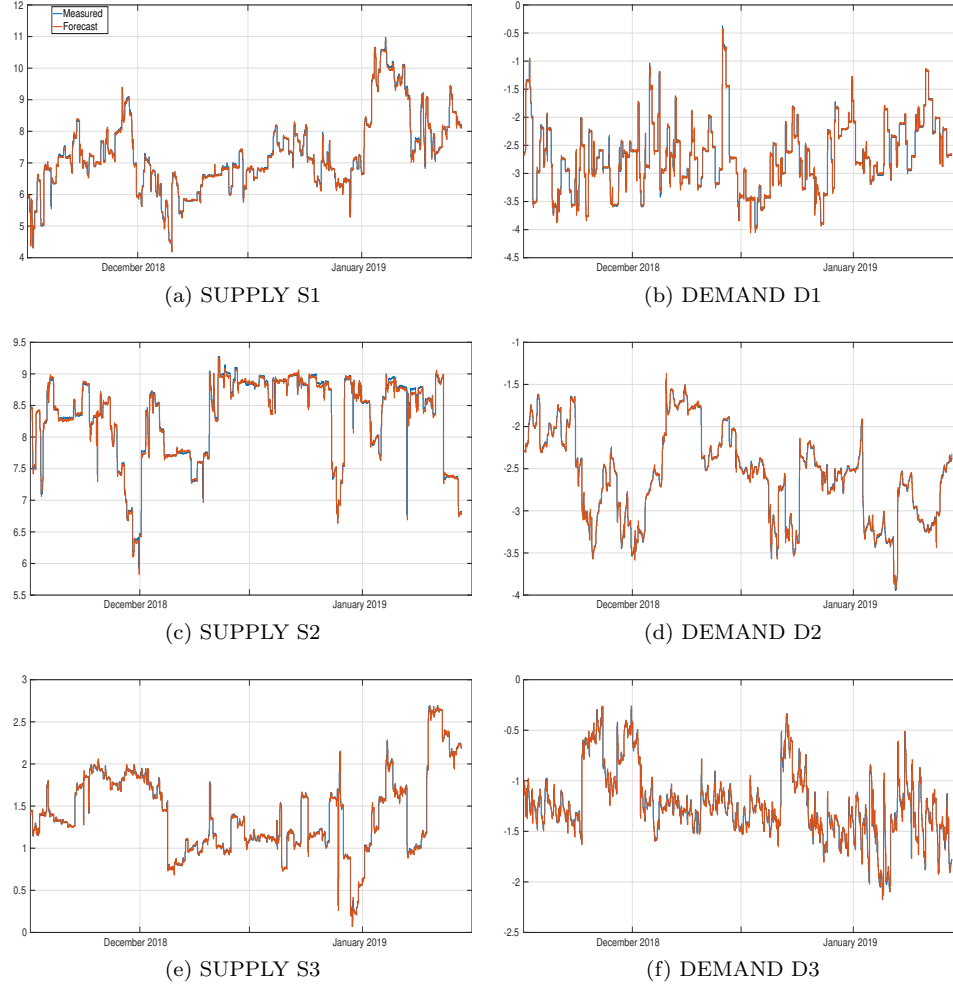


Figure 9: Measured and forecasted flow for 3 biggest supply and demand nodes

References

- [1] International Energy Agency. Germany 2020, Energy policy review. Technical report, International Energy Agency, 2020.
- [2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [3] Haydar Aras and Nil Aras. Forecasting residential natural gas demand.

Energy Sources, 26(05):463–472, 2004.

- [4] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.
- [5] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 273–283, 2018.
- [6] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. Autoregressive convolutional neural networks for asynchronous time series. In *ICML*, 2018.
- [7] Anastasia Borovykh, Sander Bohte, and Cornelis W. Oosterlee. Conditional time series forecasting with convolutional neural networks, 2017.
- [8] Marek Brabec, Ondrej Konár, Emil Pelikán, and Marek Malý. A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers. *International Journal of Forecasting*, 24(4):659–678, 2008.
- [9] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- [10] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs. *arXiv e-prints*, page arXiv:1312.6203, Dec 2013.
- [11] Bundesnetzagentur. Monitoring report 2019. Technical report, BUNDESKARTELLAMT, 2019.
- [12] Ying Chen, Wee Song Chua, and Thorsten Koch. Forecasting day-ahead high-resolution natural-gas demand and supply in Germany. *Applied Energy*, 228:1091 – 1110, 2018.
- [13] Michaël Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, 2016.
- [14] Erkan Erdogdu. Natural gas demand in Turkey. *Applied Energy*, 87(1):211–219, 2010.
- [15] Krauss Fisher. Deep learning with long short-term memory networks for financial market predictions. *FAU Discussion papers in Economics*, 2017.
- [16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 1243–1252. JMLR.org, 2017.

- [17] Open Grid Europe GmbH. <https://www.oge.net/>.
- [18] Fatma B. Gorucu. Evaluation and forecasting of gas consumption by statistical analysis. *Energy Sources*, 26(3):267–276, 2004.
- [19] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):922–929, 2019.
- [20] D.K. Hammond, P.Vanderghelynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.
- [21] Felix Hennings, Lowis Anderson, Kai Hoppmann-Baum, Mark Turner, and Thorsten Koch. Controlling transient gas flow in real-world pipeline intersection areas. *Optimization and Engineering*, 2020.
- [22] J.H. Herbert. An analysis of monthly sales of natural gas to residential customers in the United States. *Energy Systems and Policy*, 10:127–147, 1987.
- [23] J.H. Herbert, S. Sitzler, and Y. Eades-Pryor. A statistical evaluation of aggregate monthly industrial demand for natural gas in the USA. *Energy Systems and Policy*, 12(12):1233–1238, 1987.
- [24] Ashesh Jain, A. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5308–5317, 2016.
- [25] Yusuf Karadede, Gultekin Ozdemir, and Erdal Aydemir. Breeder hybrid algorithm approach for natural gas demand forecasting model. *Energy*, 141(C):1269–1284, 2017.
- [26] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182:72 – 81, 2019.
- [27] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR '17*, 2017.
- [28] R. Kizilaslan and B. Karlik. Comparison neural networks models or short term forecasting of natural gas consumption in Istanbul. *Applications of Digital Information and Web Technologies. ICADIWT*, pages 448–453, 2008.
- [29] R. Kizilaslan and B. Karlik. Combination of neural networks forecasters for monthly natural gas consumption prediction. *Neural Network World*, pages 191–199, 2009.

- [30] Lara-Benitez, P. Carranza-García, and J.M. M. Luna-Romera. Temporal convolutional networks applied to energy-related time series forecasting. *Appl. Sci.*, 10:2322, 2020.
- [31] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2017.
- [32] Lon-Mu Liu and Maw-Wen Lin. Forecasting residential consumption of natural gas using monthly and quarterly time series. *International Journal of Forecasting*, 7(1):3–16, May 1991.
- [33] S.A. Nabavi, A. Aslani, M.A. Zaidan, M. Zandi, S. Mohammadi, and N. Hossein Motlagh. Machine learning modeling for energy consumption of residential and commercial sectors. *Energies*, pages 208 – 220, 2020.
- [34] Working Group on Energy Balances. Evaluation tables on the energy balance for the federal republic of germany 1990 to 2019. Technical report, Arbeitsgemeinschaft Energiebilanzen, 2019.
- [35] Ioannis P. Panapakidis and Athanasios S. Dagoumas. Day-ahead natural gas demand forecasting based on the combination of wavelet transform and anfis/genetic algorithm/neural network model. *Energy*, 118:231 – 245, 2017.
- [36] L. Sehovac and K. Grolinger. Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention. *IEEE Access*, 8:36411–36426, 2020.
- [37] Pramono SH, Rohmatillah M, Maulana E, Hasanah RN, and Hario F. Deep learning-based short-term load forecasting for supporting demand response program in hybrid energy system. *Energies*, 12(17):3359, 2019.
- [38] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1616–1626, 2018.
- [39] Jolanta Szoplik. Forecasting of natural gas consumption with artificial neural networks. *Energy*, 85:208 – 220, 2015.
- [40] Reed P. Timmer and Peter J. Lamb. Relations between temperature and residential natural gas consumption in the central and eastern united states. *Journal of Applied Meteorology and Climatology*, 46(11):1193–1203, 2007.
- [41] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

- [42] C. A. Villacorta Cardoso and G. Lima Cruz. Forecasting natural gas consumption using ARIMA models and artificial neural networks. *IEEE Latin America Transactions*, 14(5):2233–2238, 2016.
- [43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *arXiv e-prints*, page arXiv:1901.00596, Jan 2019.
- [44] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [45] Huaxiu Yao, F. Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Z. Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [46] Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 6410–6421, 2018.
- [47] Ting Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *IJCAI*, 2018.