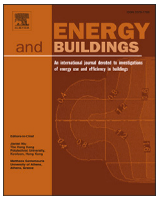
Energy & Buildings 314 (2024) 114236

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild



Benchmarking reservoir computing for residential energy demand forecasting

Karoline Brucke [a,*], Simon Schmitz [b], Daniel Köglmayr [c], Sebastian Baur [c], Christoph Räth [c], Esmail Ansari [d], Peter Klement [a]

[a] DLR-Institute of Networked Energy Systems, Carl-von-Ossietzky-Str. 15, Oldenburg, 26129, Germany
[b] DLR-Institute of Software Technology, Linder Höhe, Cologne, 51147, Germany
[c] DLR-Institute for AI Safety and Security, Wilhelm-Runge-Straße 10, Ulm, 89081, Germany
[d] Fraunhofer Institute for Manufacturing Technology and Advanced Materials IFAM, Wiener Straße 12, Bremen, 28359, Germany

ARTICLE INFO

Keywords:
Reservoir computing
Next generation reservoir computing
Recurrent network architectures
Energy demand forecasting
LSTM

ABSTRACT

In the energy sector, accurate demand forecasts are vital but often limited by the available computational power. Reservoir computing (RC) or echo-state networks excel in chaotic time series prediction, with lower computational requirements compared to other recurrent network based methods like LSTMs. Next-generation reservoir computing (NG-RC) is a newer, more efficient variant of classical RC originating from nonlinear vector autoregression and therefore missing the randomness of classical RC. In our study, we evaluate RC and NG-RC for day-ahead energy demand predictions on four data sets and compare it to LSTMs and a naive persistence approach. We find that NG-RC outperforms all other methods when considering the root mean squared error on all data sets but struggles with very small or zero demands. Additionally, it offers a very computationally effective hyperparameter optimization and excels in replicating the inherent volatility and the erratic behavior of energy demands.

1. Introduction

High quality demand predictions are crucial in energy management to e.g. make full use of flexibility potentials or enable demand response [1,2]. Especially, the heat sector offers high flexibility potential with its inherent storage properties due to the inertia of heat systems [3,4]. Managing the electricity loads on the other hand can also provide flexibility by applying prediction based demand side management like load shifting etc. [5,6]. Especially, the residential sector is of importance due to its impact in the distribution grids and its high but currently unused flexibility potential [7].

Typical approaches to carry out demand predictions in the energy research are divided into artificial intelligence (AI) based methods and non-AI based methods [8]. Non-AI based methods comprise mainly of the generation of standardized load profiles [9], statistical approaches like ARIMA [10] or naive forecasts such as persistence approaches. AI based approaches mainly focus on the use of Support Vector Regression (SVR) [11] and artificial neural networks (ANNs) such as Long Short Term Memory (LSTM) Networks [12,13]. Generally, ANNs and especially LSTMs outperform most other techniques and are often considered state-of-the-art for high quality demand prediction [14]. But they certainly have downsides regarding the high computational power and time needed in training and hyperparameter optimization as well as extensive prior feature engineering effort and the high amount of required training data.

Reservoir Computing (RC) [15,16] has been proven to perform outstandingly well in the prediction of (chaotic) time series especially on theoretical data sets like Lorenz oscillator systems [17]. It can be understood as a generalized form of echo-state networks (ESN) and sometimes these terms are used equivalently. In Reservoir Computing, only the output layer is trained while the so-called reservoir is kept fix. Therefore, the comparably low training effort makes RC especially appealing for any real world applications where large scale computation hardware might not be available [18]. Additionally, this type of recurrent neural network is known to perform well even at small amounts of training data. Scientific publications show good results in the prediction of real

[*] Corresponding author.
E-mail address: Karoline.Brucke@dlr.de (K. Brucke).

https://doi.org/10.1016/j.enbuild.2024.114236
Received 29 January 2024; Received in revised form 5 April 2024; Accepted 30 April 2024
Available online 8 May 2024

world data like stock prices [19], wind speeds [20] or solar radiation [21]. Physical reservoirs are also used as in [22] where the authors predict household energy demands with a spintronic physical reservoir. Nevertheless, the research on the application of RC or any type of echo state networks (ESN) to energy demands is not extensive:

In [23], [24] and [25] the authors use ESN to carry out a 24 h day-ahead load prediction at an hourly resolution based on utility company data. Commercial buildings are the focus of [26] and [27] developing load predictions using RC while other publications use household data when applying ESN to load forecasting like [28], [29] and [18]. Multiple publications combine classical ESN or RC with methods to intelligently preprocess the data like in [30] where the authors apply wavelet transformation and neural reconstruction or like [31] where principal component analysis is used to reduce dimensionality of the load forecast. Another research focus applying RC methods to load forecasting problems is the hyperparameter optimization of the ESN method. Particle swarm optimization is used for that task in [32] while genetic algorithms are applied in [31] and [29]. Most of the publications compare multiple RC approaches and variations against each other [27,33,29]. Only a few compare them to other types of recurrent neural networks like LSTMs as well [28,22]. Even fewer publications consider multiple data sets or different types of data: In [34] the authors use deep ESN to predict energy consumption as well as wind power generation and in [35], high order fuzzy cognitive maps are used as reservoir computing models to predict solar energy as well as load. While there are first applications of RC for probabilistic load forecasting, like the authors show in [36] by coupling the ESN with multiple methods of uncertainty, we focus on determining point forecasts in this work. In [37], the authors use ESN to predict the heating water temperature in the heat exchange station controller but to the best of our knowledge, RC has not yet been applied to (space) heat demands directly.

Recently, a new type of RC called *Next-Generation Reservoir Computing* (NG-RC) has been introduced [38]. It is characterized by the lack of randomness, the smaller number of tunable hyperparameters, and the potential computational speedup compared to classical reservoir computing. In addition to these operational advantages, it has been shown in several publications that NG-RC requires significantly less training data than the already barely data-hungry classical reservoir computing [38–40], making NG-RC a highly efficient method for predicting time series data. So far, however, the application examples of NG-RC related publications focus on dynamical model systems, rather than real system data. To the best of our knowledge, the architecture has not yet been applied in the literature to predict energy demands of any kind.

In this work, we want to demonstrate and evaluate the application of RC and NG-RC to four different energy demand data sets from the residential sector. For this purpose, we firstly conduct the RC and NG-RC based prediction of space heating demands based on simulated space heat demand data of 5 years for a whole residential district. Secondly, we apply both methods to three different sets of measured residential electric loads of two years provided by the EMSIG data set [41]. Thirdly, we investigate the sensitivity of RC and NG-RC to the amount of available training data since RC-based methods are known for delivering high quality results even at very small training sets. This investigation is done for all four data sets. All RC-based results are compared to the performance of a state-of-the-art LSTM with respect to a selected set of evaluation metrics. Finally, we benchmark everything against a naive persistence forecasts. Doing so, we can also conclude findings regarding the behavior of both approaches with respect to the type of data to be predicted. Generally, energy demand predictions are divided into *very short term*, *short term*, *medium term* and *long term* with respective cut off forecast horizons of one day, two weeks and three years [42]. Especially, the 24 h-day-ahead forecast is of importance in energy management and therefore considered in this work [43,44]. The data granularity of all data sets equals to 15 min which concludes to a prediction horizon of 96 time steps for 24 h, respectively.

The main research contributions of this paper are summarized as follows:

- First application of NG-RC on energy demand data
- Comparison between NG-RC, standard RC and state-of-the-art LSTMs on 4 independent energy demand data sets representing single households or whole districts
- The sensitivity of all methods to the amount of available training data is examined on all data sets

In the following, we start by describing all four used data sets in Section 2. Afterwards, the used methodology is explained in Section 3 by briefly describing LSTMs in Section 3.2, RC in Section 3.3, NG-RC in Section 3.4 and as well as the later used error measures to evaluate the results in Section 3.5. Thereafter, results are presented and discussed as we will comparatively evaluate the prediction results of the two reservoir computing techniques in Section 4. We conclude with a summary and outlook in Section 5.

2. Data

2.1. Simulated space heat demand data

The historical space heat demand profiles were simulated using the software QuaSi [45,46]. The buildings under study were calibrated for a standard weather profile applying simplified cubatures and determining of the building material properties, in order to comply with the annual space heat demand estimations according to the energy performance certificates of the buildings following DIN 4108 [47]. QuaSi simulates the buildings energetic behavior and thereby can create hourly or 15-minutes load profiles for space heating using a generic thermal building model based on EnergyPlus [48]. These calibrated models in QuaSi were then applied to generate the historical space heat demand profiles using the historical hourly weather data published by DWD [49] from 2017 to 2021 for the location of Bremen, Germany. The few sporadic missing weather data were closed utilizing interpolation techniques based on reasonable assumptions. As QuaSi can only process the weather data in TRY-format, the historical weather data were hence mapped to this format. The underlying building model represents a small residential district containing four apartment buildings with 108 living units and 6963 square meters of net living space.

2.2. Measured electricity demands based on the EMSIG data set

As measured electricity demands, the *EMSIG* data set is used in this work [41]. The data set represents energy data recorded by decentralized household energy management systems (EMS) from the DACH region with a granularity of 15 min. Ten different EMS are available and in this work, EMS 3, 4 and 5 are used to evaluate the different prediction methods due to their high data quality and differing characteristics. The period of time used in this work is from January 1, 2019, until December 31, 2020. The data sets represent one household each. The column `_sum_consumption_active_power` is taken for the prediction as it represents the sum of active power of the electricity consumption of the EMS. Past data is taken into account as input feature of the models while the future demands serve as output. Since every EMS represents a different building, individual models are trained on each of the data sets to carry out the predictions. This holds true for all prediction methods applied. A collection and usage of weather data as an input feature for the predictions is not applicable for the EMSIG data set since the exact locations of the buildings are not published. Detailed information regarding the properties of the measured households is not published in [41]. However, Table 1 shows selected characteristics of the 3 EMS data sets. Note, that the household that corresponds to EMS4 only has half the annual demand and peak load compared to EMS3 and EMS5. This is likely due to a smaller number of people living in the respective household.

Table 1
Selected characteristics of the data sets EMS3, EMS4 and EMS5 for the period of time from January 1, 2019, until December 31, 2020. All data sets represent individual households.

	EMS3	EMS4	EMS5
Max load [kW]	11.18	5.40	9.23
Mean load [kW]	0.73	0.34	0.79
Mean annual demand [kWh]	6307.71	2971.93	6894.36

3. Methodology

In total, four different methodologies are applied to the demand forecasting problem in this work. They include: A persistence forecast described in Section 3.1 as benchmark, state-of-the-art LSTM networks explained in Section 3.2 and classical RC as well as NG-RC which are briefly introduced in the Sections 3.3 and Section 3.4.

3.1. Persistence forecast

For the persistence forecast, past data is used to predict future demands. Since the data sets all have a granularity of 15 min and the forecast horizon is set to 24 hours, one prediction comprises 96 values. The following relationship applies to the conducted persistence prediction:

$$\hat{y}_t = y_{t-96} \tag{1}$$

This operation is carried out for all 96 time steps in one prediction while $\hat{y}_t$ represents the predicted value at time step t and y_t is the true or measured value at time step t.

3.2. Long-short-term-memory neural networks

A Long Short-Term Memory (LSTM) neural network is a type of recurrent neural network (RNN) introduced by Hochreiter and Schmidhuber which is particularly well suited for sequential data since the method overcomes the vanishing gradient problem in traditional RNNs [50]. For this purpose, LSTMs use memory cells equipped with three gates - input, output, and forget - to store and access information over extended periods of time. The gates selectively retain and discard information to handle long-term dependencies in sequential data. A schematic representation of the LSTM method is shown in the upper graph in Fig. 1. In comparison to RC and NG-RC not only the output layer but the whole LSTM network including all its trainable parameters is optimized during training. A simplified representation of an individual LSTM cell including its inputs and outputs is shown at the bottom of the respective graph. LSTMs have been successful in modeling and forecasting sequential data and are often considered state-of-the-art methodology for many applications of time series prediction. To model the LSTM networks, python and the respective `tensorflow` and `Keras` modules are used in this work [51,52].

3.3. Reservoir computing

In this work, we use a basic setup for our reservoir computing algorithm. A schematic representation of reservoir computing is shown in the middle graph of Fig. 1. The reservoir A is a sparse random network of size n with Erdős–Rényi connectivity and randomly assigned connection strengths with average node degree κ and spectral radius ρ. The input coupler $\mathbf{W}_{\text{in}}$ is a sparse random matrix with elements in [-1, 1], where each reservoir node is connected to only one element of the input. Using tanh as activation function the reservoir state $\boldsymbol{r}(t)$ can be advanced in time by one time step Δt using the previous reservoir state and the input data $\boldsymbol{x}(t)$

$$\boldsymbol{r}(t+\Delta t) = \tanh\left(\mathbf{A}\boldsymbol{r}(t) + \mathbf{W}_{\text{in}}\,\boldsymbol{x}(t)\right)\ . \tag{2}$$

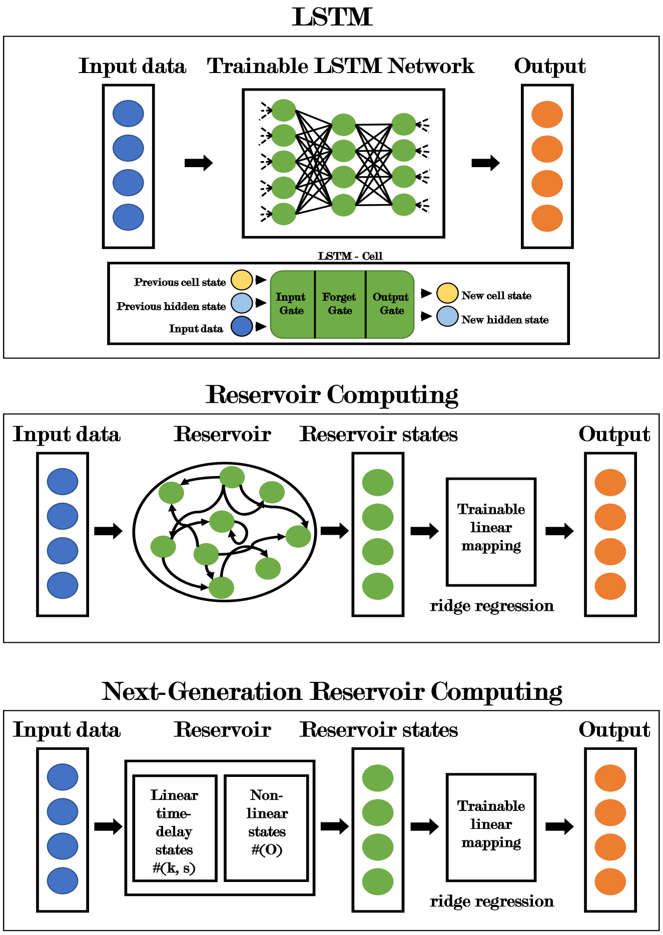


Fig. 1. Schematic representation of the used methodologies (LSTM: top, RC: middle, NG-RC: bottom).

During training $\boldsymbol{r}(t)$ is recorded and used to fit the output matrix $\mathbf{W}_{\text{out}}$ via ridge regression

$$\mathbf{W}_{\text{out}} = \arg\min_{\mathbf{W}_{\text{out}}} \left\|\mathbf{W}_{\text{out}}\tilde{\boldsymbol{r}}(t) - \boldsymbol{y}_{\text{T}}(t)\right\| + \beta\left\|\mathbf{W}_{\text{out}}\right\|\ , \tag{3}$$

where $\boldsymbol{y}_{\text{T}}(t)$ is the target output, β the regularization parameter and $\tilde{\boldsymbol{r}}$ is the quadratic Lu readout given as $\tilde{\boldsymbol{r}} = [\boldsymbol{r}, \boldsymbol{r}^2]^T$ [53]. Once trained, the output $\boldsymbol{y}(t)$ can be calculated from the reservoir state $\boldsymbol{r}(t)$ as

$$\mathbf{y}(t) = \mathbf{W}_{\text{out}}\tilde{\boldsymbol{r}}(t)\,. \tag{4}$$

When using RC for prediction, it can be run autonomously by using the prediction of the previous time step $\boldsymbol{y}(t)$ as the input $\boldsymbol{x}_{\text{pred}}(t)$ to calculate the next predicted time step $\boldsymbol{y}(t+\Delta t)$ with

$$\boldsymbol{r}(t+\Delta t) = \tanh\left(\mathbf{A}\boldsymbol{r}(t) + \mathbf{W}_{\text{in}}\,\boldsymbol{x}_{\text{pred}}(t)\right)\ . \tag{5}$$

We also embed the input data $\boldsymbol{x}(t)$ using delay embedding given by embedding parameter τ, where τ is a tuple which contains the embedding dimension and respective time delays. These parameters are optimized in a hyperparameter search, described in section 3.7.

3.4. Next generation reservoir computing

Next generation reservoir computing (NG-RC) originates from nonlinear vector autoregression (NVAR) and uses a library of unique monomials of time-shifted input variables of the input data to create its feature vector, rather than using a randomly weighted internal network as used in classical reservoir computing [38]. The created feature vector is mapped to the next data point of the time series using an output layer.

As in classical reservoir computing, only the output layer is optimized during training using ridge regression. NG-RC can often be deployed with small feature vectors, which, together with the computationally inexpensive regression optimization, results in a highly efficient algorithm. To generate the feature vector, k past time series data points, each separated by s time steps, are concatenated into a preliminary linear time-delay state vector. The final feature vector is then created by concatenating the unique monomials of orders O of the entries in the linear time-delay state vector into a new nonlinear state vector. While orders larger than 1 are the only source of nonlinearity in this setup, setting $O = 1$ reduces the next generation reservoir computing architecture so that it originates from vector autoregression (VAR). The schematic representation of NG-RC is shown in the bottom graph in Fig. 1, where the functionality of the random reservoir network from the classical RC in substituted by the linear time-delayed states and the nonlinear states of orders O.

Applying NG-RC on one-dimensional input data $x(t)$ like the energy demand data in this work, firstly embeds the data by taking k previous data points which are separated by s time steps. This can mathematically be described by applying a time delay expansion operator $\mathbf{L}_k^s$ on the input data, e.g.

$$\mathbf{L}_{k=2}^{s=3}(x(\tau)) = \begin{pmatrix} x(\tau) \\ x(\tau - 3\Delta t) \end{pmatrix} . \tag{6}$$

The time-delayed state is then transformed with unique monomials of specific orders O into a higher-dimensional nonlinear feature vector by applying a multiplication operator $\mathbf{P}^O$. For example, expanding the time-delayed state vector of Eq. (6) by the unique monomials of $O = 1, 2$ leads to:

$$\mathbf{P}^{O=[1,2]}(\begin{pmatrix} x(\tau) \\ x(\tau - 3\Delta t) \end{pmatrix}) = \begin{pmatrix} x(\tau) \\ x(\tau - 3\Delta t) \\ x(\tau)^2 \\ x(\tau - 3\Delta t)^2 \\ x(\tau)x(\tau - 3\Delta t) \end{pmatrix} \tag{7}$$

This way the NG-RC reservoir vectors $\mathbf{r}(t)$ are produced with $\mathbf{r}(t) = \mathbf{P}^O(\mathbf{L}_k^s(x(t)))$.

These vectors are mapped to the next time point by matrix multiplication to perform the prediction. During training, this mapping is optimized using ridge regression, which generates the readout matrix $\mathbf{W}_{out}$. This is identical to the training procedure in traditional reservoir computing. The prediction is performed with

$$x(t + \Delta t) \approx \mathbf{W}_{out}\mathbf{r}(t) \tag{8}$$

and can recursively be applied.

3.5. Evaluation metrics

Three error measures are used in this work to evaluate the prediction results of all three methods. They are calculated for the test data sets of both data sources (electricity demand and space heat demand) and will be the same for all methods respectively. The metrics include the root mean squared error (RMSE) and the mean absolute error (MAE) as absolute error measures and the mean absolute percentage error (MAPE) as a relative error measure. They are presented in the following Equations (9), (10) and (11).

$$RMSE(y, \hat{y}) = \sqrt{\frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{T}} \tag{9}$$

$$MAE(y, \hat{y}) = \frac{\sum_{t=1}^{T} |y_t - \hat{y}_t|}{T} \tag{10}$$

$$MAPE(y, \hat{y}) = \frac{100\%}{T} \sum_{t=1}^{T} |\frac{y_t - \hat{y}_t}{y_t}| \tag{11}$$

In scientific publications investigating reservoir computing on theoretic data sets also attractor metrics like the Lyapunov exponent are considered [17]. They show how well the general nature and environment of the data are reproduced long term by the prediction and if the system is diverging. Looking at the long term behavior of the prediction methods beyond the forecast horizon of 24 hours is not critical in energy management. Therefore, no consideration of these attractor error measures is given in this work. The given test set consists of N time steps. A prediction is carried our for every point in time $n \in \{1...N\}$ of the test set for the following 24 hours, thus 96 time steps. This results in $N - 96$ single predictions. Every prediction is then evaluated by calculating all presented error measures. Subsequently, the mean of all errors is calculated which will be referred to as the total error on the test set.

3.6. Data preprocessing

Space heat demand data. The data set of simulated space heat demands contains data from January 1, 2017, until December 31, 2021, which is - aligned with best practice - split into training set, validation set and test set for the LSTM models. Since space heat demands show a massive seasonal dependency, all data sets have to contain data from all seasons. Therefore, data from 2017, 2018 and 2019 is used for training. Data from 2020 is used for validation during training and hyperparameter optimization and 2021 is used for testing. The test set is kept same for all methods to ensure comparability of the results. RC and NG-RC do not need a validation set, so they are trained on data from 2017 to 2020 and tested on 2021, respectively. During the summer time, space heat demands are usually zero since no domestic hot water demands are included in the depicted space heat demand data.

Main input features: Past space heat demands are the most important feature in training the prediction models. RC and NG-RC use this as the only input feature for their predictions.

Additional features: For the LSTM only, calendrical features and weather features are used in addition to past space heat demands. The time features comprise an integer as flag for weekdays, weekends and holidays as well as an integer for the time of day. As weather features the weather data that was used to generate the space heat demand profiles is also used as input feature for the prediction available in hourly resolution from [49]. It gets re-sampled to a granularity of 15 min by forward fill. The respective measuring station ID is 691 which corresponds to the location Bremen, Germany. The used weather features are diffuse and global radiation, wind speed, wind direction, temperature, relative humidity, air pressure and degree of cloud coverage. In total this corresponds to 12 additional input features for the LSTM and 1 output feature being the predicted demand for the next 96 time steps. One input sample therefore contains 13 columns and 96 rows for each time step in the sample since the last 24 h of data is used to predict the following 24 h. RC and NG-RC do not use these additional features as they didn't improve their performance.

Normalizing: For the LSTM, all features are scaled individually by the `MinMaxScaler` provided by `scikit-learn` to a range of $[0, 1]$ [54]. For RC the symmetry of data with regards to zero is important. Thus, the input data is normalized to a mean of zero and standard deviation of 1 and smoothed by convolution with a kernel width of 5. NG-RC does not use an activation function in the classical sense which is why scaling of input data is not necessary. Note, this procedure is performed for all data sets.

Electricity demand data. The electricity demand data is prepared and processed the same way as the space heat demand data despite two differences: Firstly, no weather data is available for the used EMS since their location is not known. Secondly, the data split into training, validation and test set for the LSTM models differs: The first 70% of data points are used for training, the following 20% are used as validation set and the remaining 10% of data is used for testing the different methods. RC and NG-RC are trained on the first 90% of data points and

Table 2
Hyperparameters of RC with respect to all applied datasets with EMS3, EMS4 and EMS5 corresponding to measured electricity demands.

RC Hyperparameter	Space heat demand	EMS3	EMS4	EMS5
n	2000	2000	2000	2000
κ	3	3	3	3
ρ	0.3	0.3	0.9	0.5
τ	(0, 48, 96)	(0, 48, 96)	(0, 96, 192)	(0, 96, 192)
β	10^{-8}	10^{-2}	10^{-2}	10^{-2}
σ	0.1	0.03	0.1	0.01

tested on the remaining 10% which are the same as for the LSTM. Here, a data split mid year is feasible, since the electricity demands show no significant seasonal dependency likely due to a non-electrified heating technology. As for the space heat demand data set, RC and NG-RC use past demands as their only input features while the LSTM additionally gets calendrical features.

3.7. Hyperparameter optimization

Persistence. The persistence forecast has neither parameters which are optimized during training nor hyperparameters to be optimized additionally.

LSTM. For every of the four data sets a hyperparameter optimization is carried out for the LSTM based prediction, respectively. This results in individually optimized LSTMs for every data set. The optimization is based on the `optuna` python module using the provided Tree-structured Parzen Estimator (TPE) as sampling algorithm [55]. The score to evaluate a LSTM hyperparameter configuration during optimization is set to the MAE calculated for the test data set.

The following hyperparameters are considered during the optimization with `optuna`: Number of LSTM layers, number of dense layers, number of neurons per layer, learning rate, batch size, beta1, beta2, epsilon, weight decay, clipnorm, clipvalue and ema momentum. An explanation of the individual impact and functionality of the hyper parameters can be found in the respective `keras` module documentation [52].

For all optimization runs, the activation function of all layers is the hyperbolic tangent. During training, the MAE is used to quantify the LSTM's performance on the training set and the validation set. For the optimization, 1000 different trials are run with different hyperparameter configurations. An early stopping is used during training to avoid over-fitting on the training data set taking into account the MAE on the validation set.

Reservoir computing. In order to optimize the hyperparameters of the RC algorithm, a grid search was performed for the following hyperparameters: Reservoir dimension n, average node degree κ, spectral radius ρ, time delay and embedding dimension τ, regularization parameter β. Additionally, a random noise with mean 0 and standard deviation σ was added to the training data resulting in increased robustness of the algorithm. Table 2 shows the optimized values of the respective RC hyperparameters for all data sets.

Next generation reservoir computing. Since the NG-RC architecture does not involve randomness, the hyperparameter optimization can be performed by simple grid search. Moreover, these are few in comparison to RC and LSTM. The hyperparameters comprise the k value, which indicates the number of past data points of the time series considered for the prediction and s being the index distance between those data points. The hyperparameter O indicates the orders involved in the nonlinear states. The results presented here were produced with the orders involved in the NG-RC set to $O = 1$. We searched extensively for orders up to order 4, with k values up to $k = 6$ and 10 different s values up to $s = 96$ and 13 different regression parameters between 10^2 and 10^{-10}. Nevertheless, the results with order $O = 1$ outperformed those with orders larger than one. This reduces the NG-RC architecture to a linear architecture with respect to the Vector Autoregression (VAR) procedure. Table 3 presents the values for the hyperparameters s and k for all applied datasets. The regression parameter was set to $\beta = 9 \cdot 10^{-10}$ for all tests.

Table 3
Hyperparameters of NG-RC with respect to all applied datasets with EMS3, EMS4 and EMS5 corresponding to measured electricity demands.

NG-RC Hyperparameter	Space heat demand	EMS3	EMS4	EMS5
k	299	180	250	280
s	1	5	5	5

4. Results and discussion

In the following, the performance of RC, NG-RC and LSTM on the prediction problem of all 4 data sets is presented. All results will be benchmarked against the simple persistence approach as described in Section 3.1. Firstly, the focus is set on the results on the space heat demand data set. Secondly, the results on the three electricity demand data sets are described and discussed. In the end, the sensitivity of all methods regarding the size of the available training set is examined using all four data sets.

4.1. Simulated space heat demands

Generally, persistence forecasts perform very well on simulated data since it contains less randomness and less complex patterns than real-world measured data. More complex methods have a hard time delivering a significant improvement over naive approaches here. This can be confirmed by the error evaluation shown in Table 4: Considering the MAE, LSTM performed best but the relative improvement over the persistence forecast is only 16%. However, looking at the RMSE, NG-RC delivered best results with a significant relative improvement of 45% over the persistence forecast. This is due to the persistence forecast reproducing the demand peaks and spikes of the day before, the RMSE is high with respect to the MAE since the RMSE penalizes outliers more than the MAE. NG-RC seems to be able to predict the peaks better than any other method. Nevertheless, none of the methods outperformed the persistence forecast considering the MAPE.

Forecasts in the winter time and summer time differ in quality. While the complex methods outperform persistence based predictions during winter times, where the demand is high, the methods have difficulties to predict exact zero demands during summer. This behavior can be seen in the upper left graph of Fig. 2 where the error measure MAE is shown over time for the whole space heat demand test set.

The distribution of errors is depicted in Fig. 3 where the MAE on the space heat demand test sets is shown as histograms. The persistence forecast has the broadest error distribution which means the most amount of outliers in the predictions which is confirmed by the high RMSE compared to MAE for persistence forecast results. On the other hand, the histograms of LSTM and RC are very similar with a peak at very small MAEs and then a normal distribution of MAEs with a mean at approximately 2 kWh for RC and 1.5 kWh for the LSTM. The behavior of NG-RC during summer can also be seen in the respective MAE histogram: The peak of very small MAEs is stretched out compared to the LSTM and RC but the second peak of the histograms is at smaller MAEs of approximately 1 kWh.

Looking at the depicted example days for all prediction methods in Fig. 4, significant differences become visible: The upper example day (January 22, 2020) shows a day where all methods perform well. On that day, the LSTM and RC predict a smooth demand pattern without any spikes. NG-RC is the only method that is able to reproduce the

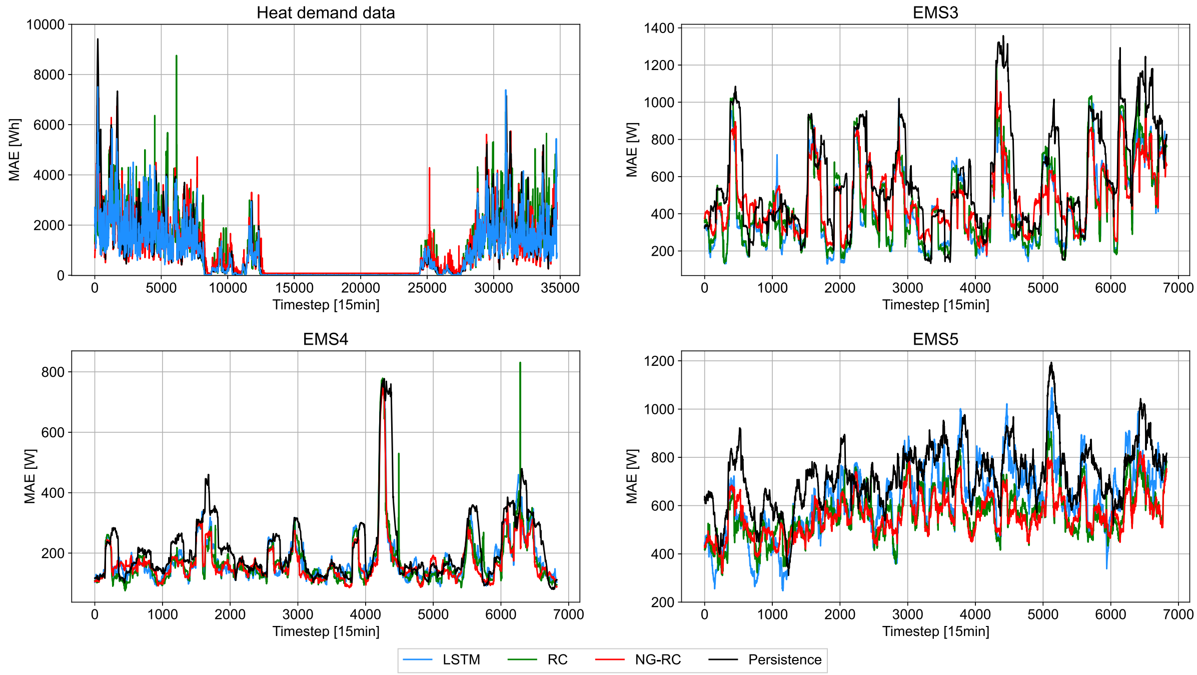


Fig. 2. MAEs for all methods and all data sets over the course of the respective test data set.

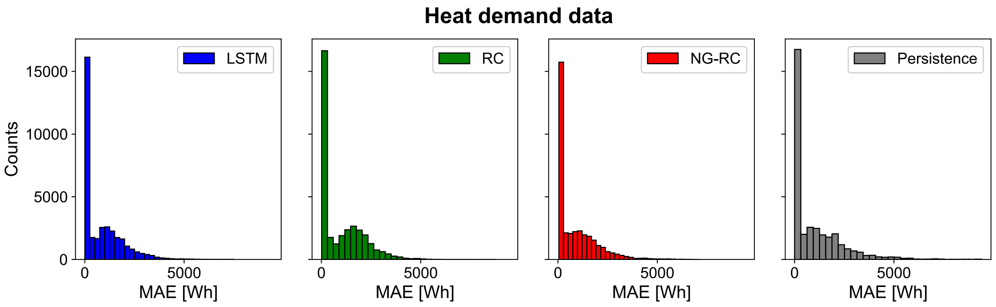


Fig. 3. Histograms of MAEs on the test set of simulated space heat demand data for all applied forecasting methods.

Table 4
Error measures of all prediction methods calculated for the space heat demand test data set with best error values in bold.

Space heat demand test set			
Prediction Method	MAE [Wh]	RMSE [Wh]	MAPE [%]
Persistence	984.40	1933.62	**23.93**
LSTM	**829.64**	1636.95	42.49
RC	938.98	1240.42	59.92
NG-RC	876.67	**1060.68**	47.11

small space heat demand peaks in the morning. These spiky predictions are also reflected in the smallest RMSE of all methods in Table 4. This could be a big advantage over other methods, since it is typical for energy consumption to be very erratic. At the same time hardly any other sophisticated method is able to predict that kind of behavior.

The example day July 20, 2020 in the middle of Fig. 4 shows a summer day without any demand. Therefore, the persistence forecast performs best, predicting only zeros due to the day before also having no demand at all. The LSTM predicts values very close to zero. RC and especially NG-RC are not able to predict values even close to zero. RC shows a noisy prediction over the course of the whole day, while NG-RC predicts an increasing demand over time. That is a main driver for the higher MAEs of these methods compared to the LSTM. This is likely the result of neither RC nor NG-RC using the additional calendrical and weather features for their predictions, hence missing some important context which would allow for such qualitatively different predictions between the seasons. Yet, for our relatively simple RC and NG-RC setups, adding these additional features to the input did not improve their prediction quality overall. This is likely due to the fact that on average this data "confuses" the models more during the other seasons than it helps them during summer. For such strongly seasonal data, it may therefore be preferable to train seasonal RC and NG-RC models to take full advantage of all available data. The last example day (October 25, 2020) shown on the bottom of Fig. 4 is a day in fall where all methods underestimate the real demands. But again, only NG-RC is able to predict the ramping behavior in the morning and performs best compared to all other methods. LSTM and RC show predictions which look like a rolling mean of previous days. During the night hours, the true demand is zero but again, especially NG-RC and RC have difficulties to predict that and also predict values smaller than zero which does not make sense for demand data. Therefore, a postprocessing of the predictions allowing only positive values would be beneficial.

4.2. Measured electricity demands

On real world data, more complex methods usually outperform simpler approaches like persistence predictions. We show the same behavior in our experiments when carrying out forecasts on real world electricity demands. In Table 5 it gets clear, that LSTM, RC and NG-RC outperform the persistence approach on all three data sets and with re-

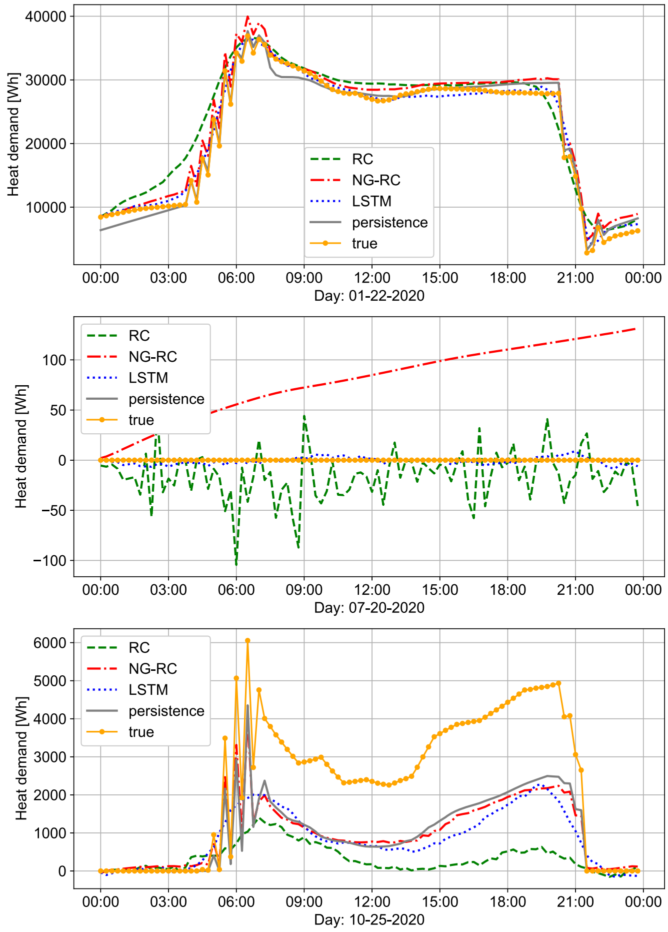


Fig. 4. Three exemplary days of the space heat demand test data set.

Table 5
Error measures of all prediction methods calculated for every test set of the used EMS from the EMSIG data set with best error values in bold.

EMSIG test data sets			
Prediction Method	MAE [W]	RMSE [W]	MAPE [%]
	EMS 3		
Persistence	588.25	1196.70	68.83
LSTM	**436.39**	933.86	**35.12**
RC	448.62	822.10	42.46
NG-RC	475.11	**783.87**	51.95
	EMS 4		
Persistence	217.56	377.32	101.06
LSTM	178.23	304.20	**62.08**
RC	172.03	265.06	62.09
NG-RC	**169.76**	**254.67**	70.19
	EMS 5		
Persistence	724.97	976.72	111.89
LSTM	621.91	868.33	**68.41**
RC	562.56	736.72	86.35
NG-RC	**560.56**	**714.13**	90.56

spect to all applied evaluation metrics. On all three EMSIG data sets, the LSTM results in the best MAPE while NG-RC has the smallest RMSE. Only on the EMS3 data set, the LSTM outperforms the NG-RC approach with respect to the MAE. These observations lead to the conclusion that NG-RC is able to perform forecasts with significantly less outliers compared to RC and LSTM. The small MAPE of the LSTM is due to its ability to predict smaller values better which are more emphasized by the MAPE compared to the other absolute error measures. The MAPE is generally higher compared to the space heat demands which is due to the higher randomness in electricity demands which cannot be predicted by any prediction method. The high granularity of 15 min also leads to more peaks in the demand patterns.

We observe no dependency of the MAE on the season when looking at the MAE over the course of all three test sets in Fig. 2. This is likely due to a non-electrified heating technology in the buildings and no air conditioning in the summer which is generally not common in North or Central Europe. When considering the MAE histograms of all methods with respect to all data sets in Fig. 5 the persistence forecast has the broadest histograms of all methods. This can also be confirmed looking at the example days of all data sets in Fig. 6.

Persistence predictions create a lot of outliers by reproducing the peaks of the previous day while all other methods predict a more smooth demand pattern. NG-RC shows the most deviations of demands and therefore reflects the true demand patterns best. Additionally, NG-RC shows the narrowest histograms for EMS 4 and EMS 5. For EMS 3, LSTM, RC and NG-RC predictions result in very similar histograms with LSTM being slightly better than RC and NG-RC since it has a higher peak at the very small MAEs. In Fig. 6 the predictions of a random day (December, 9, 2020) are shown for EMS 3, 4 and 5 respectively. The different characteristics of the independent data sets are clearly visible. The differences of the predicted demand patterns are similar to the ones on the space heat demand data set from the previous section.

Sensitivity to training set size. To investigate the sensitivity of each method with respect to the amount of available training data, the size of the training data set is systematically reduced, while the test set always remains the same. It corresponds to the test set of the other experiments to ensure comparability. The reduction of the training data is done by halving the non-test data. Only the second half is used for training and validation. This procedure is repeated until the non-test data set contains only about two weeks of data. This is true for all four data sets. Accordingly, one more reduction step is possible for the space heat demand data because it contains five years of data compared to two years of data in the electricity demand data sets.

For the LSTM, the respective non-test data set is split into 80% training and 20% validation data. RC and NG-RC do not require a validation set, but RC uses the first 100 time steps of the training data for its synchronization phase and NG-RC uses the first ks time steps for its warmup time. The synchronization phase and the warmup time is not considered in the error evaluation. All procedures are trained individually on each of the reduced training data sets. For RC and NG-RC, new hyperparameters are optimized for each reduction. Due to the inherent randomness of RC, 1000 reservoirs are created for each reduction and trained individually. The results of the best reservoir are considered here. The MAE is calculated as the evaluation measure. The MAE of the persistence prediction is shown as a black benchmark line and is the same for all training set sizes.

Fig. 7 shows the MAE of all methods for all training set sizes for the four data sets: For the simulated space heat demand data set, a clear dependence of the MAE of LSTM and RC on the training set size is observed. Especially the LSTM MAE increases almost exponentially after the second training set size reduction. LSTM yields higher MAEs than the persistence forecast after the second reduction, RC after the third reduction and NG-RC after the fourth reduction. Consequently, NG-RC shows the highest robustness regarding the decrease of amount of training data with a relative MAE increase of only approximately 36% when reducing the amount of training data by 99.3% from first to last reduction. This is consistent with the reported results of NG-RC requiring only little training data, although the architecture used here originates from the VAR approach by only using order $O = 1$ [38–40].

For the electricity demand data sets EMS3, EMS4 and EMS5, the dependence of RC, NG-RC and also LSTM on the size of the training set

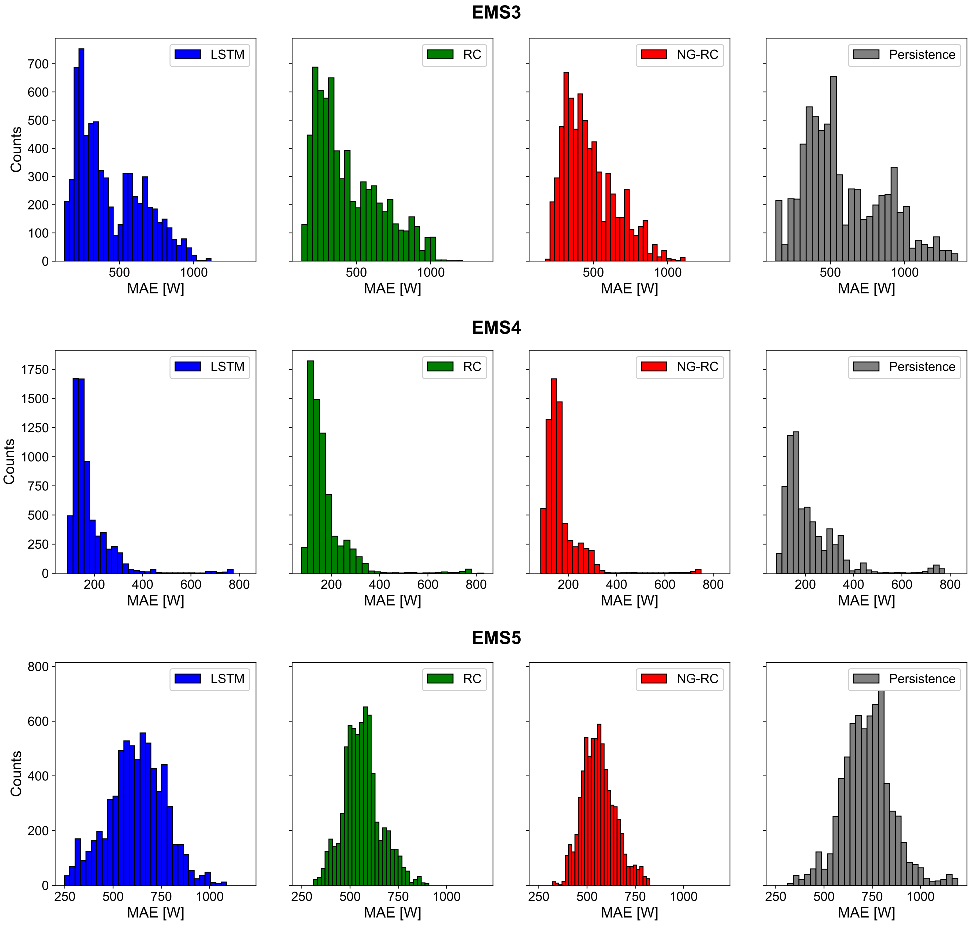


Fig. 5. Histograms of MAEs on the test set of all electricity demand data sets for all applied forecasting methods.

is not as significant as for the space heat demands. All methods show a small increase in MAE for smaller test data sets, with this increase being smallest for RC for all data sets. Nevertheless, all methods are able to outperform the persistence prediction even for the smallest training sets.

5. Conclusion and outlook

In this work, we compared three different forecasting methodologies, namely RC, NG-RC and LSTMs on four different energy demand data sets for a 24 h energy demand prediction with a granularity of 15 min. All results are benchmarked against a naive persistence forecast. One data set contains simulated space heat demand data of a residential district while the three other data sets provide measured household electricity demands taken from the *EMSIG* data set. On all data sets, the methods LSTM, RC and NG-RC are able to create better forecasts than the persistence prediction. The increase of forecast quality is highest for NG-RC considering the RMSE as evaluation metric with 27%-45% improvement with respect to persistence predictions and different data sets. Generally, we show the first application of NG-RC on energy demands. Considering the MAE, results are comparably good as for classical RC and LSTMs but the required training and optimization time is significantly smaller compared to classical RC and orders of magnitude smaller compared to LSTMs. Additionally, NG-RC is able to predict demand patterns with more peaks and deviations which is closer to real world behavior and also yields the significantly smallest RMSEs on all data sets under consideration. We also examined the sensitivity of all methods on the amount of provided training data. For space heat demands, NG-RC shows the highest robustness regarding the reduction of training data while RC showed the least increase of prediction error for the electricity demands.

As NG-RC has been successfully applied to various energy demand datasets here, future work should investigate its application to real-world prediction problems in more detail. In particular, NG-RC's ability to predict demand peaks makes it interesting for data types that exhibit irregular behavior, as in the energy sector. Furthermore, additional pre- and post-processing of the input data could ensure a better consistency with the given physical conditions. Due to the seasonal fluctuations in the prediction performance, the overall performance could be significantly increased using multiple seasonal models. Moreover, hybrid reservoir computing methods show promising approaches for incorporating known functionalities into the training [56,57]. For example, daily fluctuations, seasonal fluctuations or even persistence could be included in the training, which can be expected to increase prediction performance even under changing conditions. Finally, the newly developed minimal reservoir computing technique [58] shows great results in its forecast horizon for chaotic systems compared to NG-RC and RC. It has yet to be applied to real data and may be another promising candidate in the field of energy demand forecasting.

Funding

This research was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) (project number 03SBE111) and the Federal Ministry of Education and Research (BMBF) in the project ENaQ (project number 03SBE111) as well as by the Helmholtz Association's Initiative and Networking Fund (INF) under the Helmholtz AI platform grant agreement (ID ZT-I-PF-5-1), Local Unit 'Munich Unit @Aeronautics, Space and Transport (MASTr).'

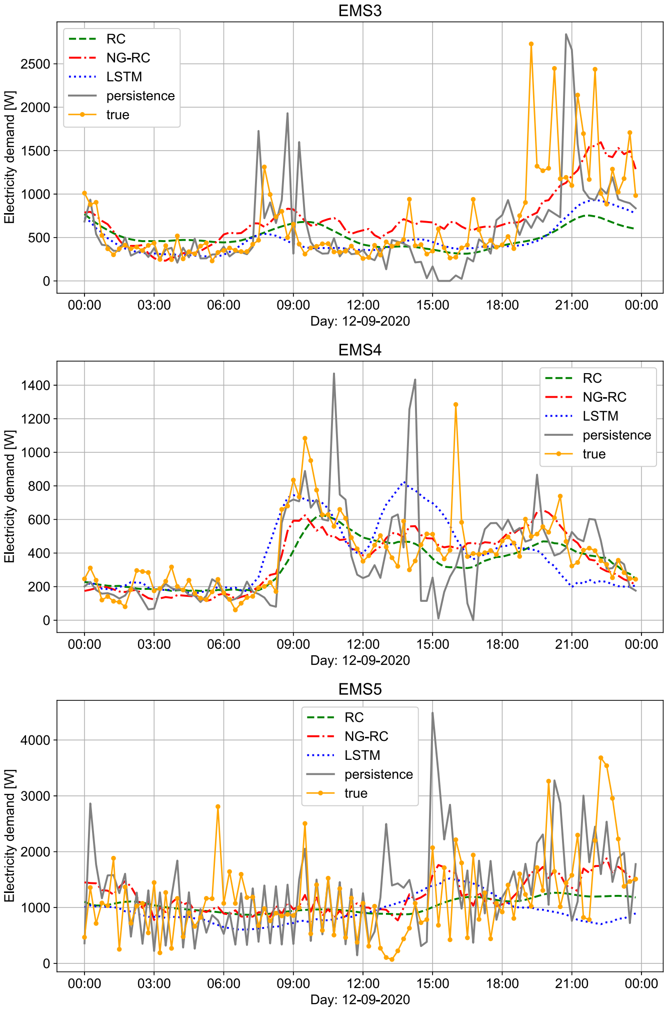


Fig. 6. Same exemplary day from every EMS test data set.

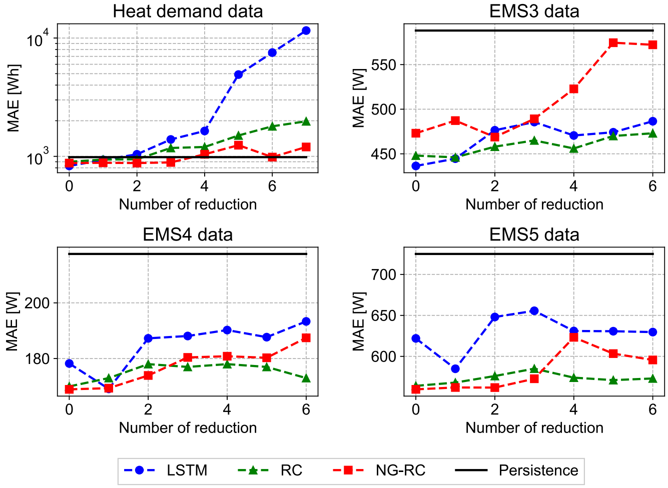


Fig. 7. MAEs of all forecasting methods on the test set of all data sets dependent on the amount of available training data. The amount of training data is halved with every reduction.

CRediT authorship contribution statement

Karoline Brucke: Writing – review & editing, Writing – original draft, Visualization, Software, Project administration, Data curation, Conceptualization. **Simon Schmitz:** Visualization, Software, Formal analysis, Data curation. **Daniel Köglmayr:** Writing – original draft, Methodology, Formal analysis. **Sebastian Baur:** Writing – original draft, Methodology, Formal analysis. **Christoph Räth:** Supervision, Methodology. **Esmail Ansari:** Writing – original draft, Data curation. **Peter Klement:** Supervision, Funding acquisition.

Declaration of competing interest

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

Data availability

We used three publicly available data sets which are referenced in the paper and one confidential data set.

References

[1] K. Amasyali, N.M. El-Gohary, A review of data-driven building energy consumption prediction studies, Renew. Sustain. Energy Rev. 81 (2018) 1192–1205.

[2] C. Deb, F. Zhang, J. Yang, S.E. Lee, K.W. Shah, A review on time series forecasting techniques for building energy consumption, Renew. Sustain. Energy Rev. 74 (2017) 902–924.

[3] J. Garcet, R. De Meulenaere, J. Blondeau, Enabling flexible CHP operation for grid support by exploiting the DHN thermal inertia, Appl. Energy 316 (2022) 119056.

[4] Y. Zhang, P. Johansson, A.S. Kalagasidis, Feasibilities of utilizing thermal inertia of district heating networks to improve system flexibility, Appl. Therm. Eng. 213 (2022) 118813.

[5] O.M. Babatunde, J.L. Munda, Y. Hamam, Power system flexibility: a review, Energy Rep. 6 (2020) 101–106.

[6] A.S. Gazafroudi, F. Prieto-Castrillo, T. Pinto, J.M. Corchado, Energy flexibility management in power distribution systems: decentralized approach, in: 2018 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, Seville, Spain, 2018, pp. 1–6.

[7] H. Golmohamadi, Demand-side flexibility in power systems: a survey of residential, industrial, commercial, and agricultural sectors, Sustainability 14 (13) (2022) 7916.

[8] M. Khalil, A.S. McGough, Z. Pourmirza, M. Pazhoohesh, S. Walker, Machine learning, deep learning and statistical analysis for forecasting building energy consumption—a systematic review, Eng. Appl. Artif. Intell. 115 (2022) 105287.

[9] N. Pflugradt, P. Stenzel, L. Kotzur, D. Stolten, LoadProfileGenerator: an agent-based behavior simulation for generating residential load profiles, J. Open Sour. Softw. 7 (2022) 3574.

[10] L.C.M. de Andrade, I.N. da Silva, Very short-term load forecasting based on ARIMA model and intelligent systems, in: 2009 15th International Conference on Intelligent System Applications to Power Systems, IEEE, Curitiba, Brazil, 2009, pp. 1–6.

[11] Y.-C. Guo, D.-X. Niu, Y.-X. Chen, Support vector machine model in electricity load forecasting, in: 2006 International Conference on Machine Learning and Cybernetics, IEEE, Dalian, China, 2006, pp. 2892–2896.

[12] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on LSTM recurrent neural network, IEEE Trans. Smart Grid 10 (1) (2017) 841–851.

[13] K. Brucke, S. Arens, J.-S. Telle, T. Steens, B. Hanke, K. von Maydell, C. Agert, A non-intrusive load monitoring approach for very short-term power predictions in commercial buildings, Appl. Energy 292 (2021) 116860.

[14] T. Hossen, A.S. Nair, R.A. Chinnathambi, P. Ranganathan, Residential load forecasting using deep neural networks (DNN), in: 2018 North American Power Symposium (NAPS), IEEE, Fargo, United States, 2018, pp. 1–5.

[15] H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80.

[16] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: a new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.

[17] J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, chaos: an interdisciplinary, J. Nonlinear Sci. 27 (12) (2017).

[18] Y. Fujimoto, M. Fujita, Y. Hayashi, Deep reservoir architecture for short-term residential load forecasting: an online learning scheme for edge computing, Appl. Energy 298 (2021) 117176.

[19] W.-J. Wang, Y. Tang, J. Xiong, Y.-C. Zhang, Stock market index prediction based on reservoir computing models, Expert Syst. Appl. 178 (2021) 115022.

[20] A.A. Ferreira, T.B. Ludermir, R.R. de Aquino, M.M. Lira, O.N. Neto, Investigating the use of reservoir computing for forecasting the hourly wind speed in short-term, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, Hong Kong, China, 2008, pp. 1649–1656.

[21] S. Basterrech, Geometric particle swarm optimization and reservoir computing for solar power forecasting, in: Recent Advances in Soft Computing: Proceedings of the 22nd International Conference on Soft Computing (MENDEL 2016), Springer, Brno, Czech Republic, 2017, pp. 88–97.

[22] W.A. Misba, H.S. Mavikumbure, M.M. Rajib, D.L. Marino, V. Cobilean, M. Manic, J. Atulasimha, Spintronic physical reservoir for autonomous prediction and long-term household energy load forecasting, IEEE Access 11 (2023) 124725–124737, https://doi.org/10.1109/ACCESS.2023.3326414.

[23] H. Showkati, A.H. Hejazi, S. Elyasi, Short term load forecasting using echo state networks, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, Barcelona, Spain, 2010, pp. 1–5.

[24] A. Deihimi, H. Showkati, Application of echo state networks in short-term electric load forecasting, Energy 39 (1) (2012) 327–340.

[25] L. Peng, S.-X. Lv, L. Wang, Z.-Y. Wang, Effective electricity load forecasting using enhanced double-reservoir echo state network, Eng. Appl. Artif. Intell. 99 (2021) 104132.

[26] M. Mansoor, F. Grimaccia, S. Leva, M. Mussetta, Comparison of echo state network and feed-forward neural networks in electrical load forecasting for demand response programs, Math. Comput. Simul. 184 (2021) 282–293.

[27] G. Shi, D. Liu, Q. Wei, Energy consumption prediction of office buildings based on echo state networks, Neurocomputing 216 (2016) 478–488.

[28] W. Bendali, M. Boussetta, I. Saber, Y. Mourad, Households energy consumption forecasting with echo state network, in: International Conference on Digital Technologies and Applications, Springer, Fez, Morocco, 2021, pp. 1305–1315.

[29] W. Bendali, I. Saber, M. Boussetta, B. Bourachdi, Y. Mourad, Optimization of deep reservoir computing with binary genetic algorithm for multi-time horizon forecasting of power consumption, J. Eur. Syst. Autom. 55 (6) (2022) 701.

[30] A. Deihimi, O. Orang, H. Showkati, Short-term electric load and temperature forecasting using wavelet echo state networks with neural reconstruction, Energy 57 (2013) 382–401.

[31] F.M. Bianchi, E. De Santis, A. Rizzi, A. Sadeghian, Short-term electric load forecasting using echo state networks and pca decomposition, IEEE Access 3 (2015) 1931–1943.

[32] N. Sinha, C. Lucheroni, Double-reservoir deep echo state network architecture for short-term electricity demand forecasting, in: 2022 18th International Conference on the European Energy Market (EEM), IEEE, Ljubljana, Slovenia, 2022, pp. 1–6.

[33] L. Wang, H. Hu, X.-Y. Ai, H. Liu, Effective electricity energy consumption forecasting using echo state network improved by differential evolution algorithm, Energy 153 (2018) 801–815.

[34] H. Hu, L. Wang, S.-X. Lv, Forecasting energy consumption and wind power generation using deep echo state network, Renew. Energy 154 (2020) 598–613.

[35] O. Orang, P.C.d.L. Silva, F.G. Guimarães, Introducing randomized high order fuzzy cognitive maps as reservoir computing models: a case study in solar energy and load forecasting, arXiv preprint, arXiv:2201.02158, 2022.

[36] M. Guerra, S. Scardapane, F.M. Bianchi, Probabilistic load forecasting with reservoir computing, arXiv preprint, arXiv:2308.12844, 2023.

[37] L. Liu, J. Liu, Z. Chen, Z. Jiang, M. Pang, Y. Miao, Research on predictive control of energy saving for central heating based on echo state network, Energy Rep. 9 (2023) 171–181.

[38] D.J. Gauthier, E. Bollt, A. Griffith, W.A. Barbosa, Next generation reservoir computing, Nat. Commun. 12 (1) (2021) 5564.

[39] W.A. Barbosa, D.J. Gauthier, Learning spatiotemporal chaos using next-generation reservoir computing, Chaos, Int. J. Nonlinear Sci. 32 (9) (2022).

[40] A. Haluszczynski, D. Koeglmayr, C. Räth, Controlling dynamical systems to complex target states using machine learning: next-generation vs. classical reservoir computing, in: 2023 International Joint Conference on Neural Networks (IJCNN), IEEE, Queensland, Australia, 2023, pp. 1–7.

[41] D. Musikhina, J. Seidemann, S. Feilmeier, [dataset] EMSIG: energy management system data, available at https://openenergy-platform.org/dataedit/view/demand/emsig_energy_data_by_ems, 2021.

[42] T. Hong, S. Fan, Probabilistic electric load forecasting: a tutorial review, Int. J. Forecast. 32 (3) (2016) 914–938.

[43] J. Honold, P. Wimmer, C. Kandler, Potential of energy management systems in residential buildings, Energy Proc. 78 (2015) 2094–2099.

[44] M.H. Albadi, E.F. El-Saadany, A summary of demand response in electricity markets, Electr. Power Syst. Res. 78 (11) (2008) 1989–1996.

[45] T. Braunschweig, Generische Gebäudesimulation als Bestandteil der Quartier-Simulationssoftware "QuaSi"-Verbundvorhaben EnStadtEs-West: Klimaneutrales Stadtquartier Neue Weststadt Esslingen, 2020.

[46] T. Maile, H. Steinacker, M.W. Stickel, E. Ott, C. Kley, Automated generation of energy profiles for urban simulations, Energies 16 (17) (2023).

[47] DIN e.V., DIN 4108-6, Wärmeschutz und Energie-Einsparung in Gebäuden - Teil 6: Berechnung des Jahresheizwärme- und des Jahresheizenergiebedarfs, 2003.

[48] D. Crawley, C. Pedersen, L. Lawrie, F. Winkelmann, Energyplus: energy simulation program, ASHRAE J. 42 (2000) 49–56.

[49] Deutscher Wetterdienst, [dataset] open data platform of Deutscher Wetterdienst, available at https://www.dwd.de/DE/leistungen/opendata/opendata.html, 2022.

[50] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[51] M. Abadi, et al., TensorFlow: large-scale machine learning on heterogeneous systems, available at https://www.tensorflow.org/, 2015.

[52] F. Chollet, keras, available at https://github.com/fchollet/keras, 2015.

[53] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, E. Ott, Reservoir observers: model-free inference of unmeasured variables in chaotic systems, Chaos, Int. J. Nonlinear Sci. 27 (4) (2017) 041102.

[54] Pedregosa, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (Oct 2011) 2825–2830.

[55] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: a next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data, Mining, New York, United States, 2019.

[56] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B.R. Hunt, M. Girvan, E. Ott, Hybrid forecasting of chaotic processes: using machine learning in conjunction with a knowledge-based model, Chaos, Int. J. Nonlinear Sci. 28 (4) (2018) 041101.

[57] D. Duncan, C. Räth, Optimizing the combination of data-driven and model-based elements in hybrid reservoir computing, Chaos, Int. J. Nonlinear Sci. 33 (10) (2023) 103109.

[58] H. Ma, D. Prosperino, C. Räth, A novel approach to minimal reservoir computing, Sci. Rep. 13 (1) (2023) 12970.