IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Data-Driven Approach to Partitioning Net Ecosystem Exchange Using a Deep State Space Model

VIOLETA TEODORA TRIFUNOV[1,2], MAHA SHADAYDEH[1], JAKOB RUNGE[2,3], MARKUS REICHSTEIN[4,5], and JOACHIM DENZLER[1,2,4] (Member, IEEE).

[1]Friedrich Schiller University Jena, Jena, Germany
[2]German Aerospace Center (DLR), Institute for Data Science, Jena, Germany
[3]Technische Universität Berlin, 10623 Berlin, Germany
[4]Michael Stifel Center Jena for Data-Driven and Simulation Science, Jena, Germany
[5]Max Planck Institute for Biogeochemistry, Jena, Germany

Corresponding author: Violeta Teodora Trifunov (e-mail: violetateodora.trifunov@uni-jena.de).

This work was funded by the ERC Synergy Grant 2019: Understanding and Modelling of the Earth System with Machine Learning (USMILE) and by the Carl Zeiss Foundation within the scope of the program line "Breakthroughs: Exploring Intelligent Systems" for "Digitization — explore the basics, use applications". We acknowledge support by the German Research Foundation and the Open Access Publication Fund of the Thüringer Universitäts- und Landesbibliothek Jena Projekt-Nr. 433052568.

ABSTRACT Describing ecosystem carbon fluxes is essential for deepening the understanding of the Earth system. However, partitioning net ecosystem exchange (NEE), i.e. the sum of ecosystem respiration (R_{eco}) and gross primary production (GPP), into these summands is ill-posed since there can be infinitely many mathematically-valid solutions. We propose a novel data-driven approach to NEE partitioning using a deep state space model which combines the interpretability and uncertainty analysis of state space models with the ability of recurrent neural networks to learn the complex functions governing the data. We validate our proposed approach on the FLUXNET dataset. We suggest using both the past and the future of R_{eco}'s predictors for training along with the nighttime NEE (NEE_{night}) to learn a dynamical model of R_{eco}. We evaluate our nighttime R_{eco} forecasts by comparing them to the ground truth NEE_{night} and obtain the best accuracy with respect to other partitioning methods. The learned nighttime R_{eco} model is then used to forecast the daytime R_{eco} conditioning on the future observations of different predictors, i.e., global radiation, air temperature, precipitation, vapor pressure deficit, and daytime NEE (NEE_{day}). Subtracted from the NEE_{day}, these estimates yield the GPP, finalizing the partitioning. Our purely data-driven daytime R_{eco} forecasts are in line with the recent empirical partitioning studies reporting lower daytime R_{eco} than the Reichstein method, which can be attributed to the Kok effect, i.e., the plant respiration being higher at night. We conclude that our approach is a good alternative for data-driven NEE partitioning and complements other partitioning methods.

INDEX TERMS Deep state space models, net ecosystem exchange, NEE partitioning, time series forecasting

I. INTRODUCTION

The task of net ecosystem exchange (NEE) partitioning is highly relevant in environmental science, as it deepens the understanding of the underlying mechanisms constraining the ecosystem function, in the global warming context for instance [1]. NEE is a measure of the net exchange of carbon between an ecosystem and the atmosphere given by:

$$NEE = R_{eco} + GPP. \tag{1}$$

Ecological variables in Eq. (1), R_{eco} and GPP, denote the total carbon flux by respiration processes of all organisms in an ecosystem, and the gross amount of carbon uptake by photosynthesis from plants, respectively. The NEE values are measured, but the problem of obtaining either R_{eco} or GPP from those measurements is ill-posed since there can be infinitely many possible solutions contributing to the given sum. Since no photosynthesis occurs during the night, GPP nighttime values are close to zero and hence nighttime NEE

(NEE_{night}) corresponds to the nighttime R_{eco}.

Probabilistic state-space models are a rich framework for an interpretable representation of the evolution of physical processes (e.g., ecosystem respiration), and are of fundamental significance in ecological time series analysis since they can take into account latent factors and the uncertainty in the observations. Deep state space models combine the interpretability and uncertainty analysis of state space models with the ability of recurrent neural networks (RNNs) to learn the complex functions governing the data. In this paper, we propose to use a deep probabilistic state space model, namely a deep state space model DeepState [2], to partition NEE. We first learn the underlying dynamical model of R_{eco} using measured NEE_{night} data and both the past and the future of R_{eco} predictors. Once the model is learned with the assistance of those predictors, we use it to predict daytime values of R_{eco} and can then subtract the predicted values of R_{eco} from the entire NEE time series to obtain GPP. To the best of our knowledge, our approach is the first to employ a deep probabilistic model for the task of NEE partitioning.

In addition to its ability to handle missing data, another novelty of our approach is that we use both the past and the future of R_{eco}'s predictors. The candidate predictors we use for learning the dynamical model of R_{eco} and predicting its values during the daytime are: global radiation (R_g), air temperature (T_{air}), water vapor pressure deficit (VPD), precipitation (PPT) and daytime NEE (NEE_{day}). We choose the predictors contributing to the best forecast of the ground truth NEE_{night} and then use them for forecasting R_{eco} during the daytime.

A. RELATED WORK

NEE partitioning. The most widely used approach for partitioning NEE into R_{eco} and GPP is the Reichstein method [3]. It uses temperature sensitivity of NEE_{night} to predict the daytime R_{eco}. GPP is then obtained by subtracting R_{eco} from NEE. Similarly to other numerical NEE partitioning methods such as that by Lasslop et al. [4], although very useful and easy to implement, it does not consider multiple co-acting factors that modulate GPP and R_{eco}. Tramontana et al. [1] develop an approach to account for the said factors modulating these two carbon fluxes by implementing a hybrid data-driven method, $NN_{C\text{-}part}$, based on feedforward neural networks [5], that can use a comprehensive dataset of soil and micro-meteorological variables. The expert knowledge is incorporated in the algorithm by introducing a photosynthesis response based on the radiation-use efficiency concept. In contrast to our approach, however, this NEE partitioning approach has a limitation in the sense that it does not handle the missing data values, nor the missing predictors. This constrains its applicability to the FLUXNET sites with large quantities of missing data, which is often the case due to the sensitivity of the sensors used for the measurements in combination with the unfavorable weather conditions at the measuring sites. Since we use a probabilistic time series forecasting method, the missing values in our R_{eco} forecasts never occur. Moreover, we do not need to approximate daytime R_{eco} or create proxies for GPP because we use both the past and the future of the selected R_{eco} predictor variables and thereby do not incur additional approximation error.

A study by Oikawa et al. [6] analyzed several partitioning methods, and conducted an independent empirical NEE partitioning. This approach consisted of measuring ecosystem-scale fluxes of stable C isotopes via a quantum cascade laser spectrometer, and pairing those measurements with the biophysical model CANVEG to obtain GPP and R_{eco}. These two carbon fluxes were on average 10-13% lower than both Reichstein-partitioned GPP and R_{eco} and GPP and R_{eco} obtained by the gap filling [7] of NEE. The authors further state that since both of these partitioning methods use NEE_{night} to infer daytime R_{eco}, they are likely overestimating GPP and R_{eco} during the day. Moreover, they emphasize that the effect of plant respiration being higher at night, known as the Kok effect [8], possibly supports this claim. Despite our approach using NEE_{night} to estimate daytime R_{eco}, we also obtain lower R_{eco} estimates than those of the Reichstein- and $NN_{C\text{-}part}$ partitioning methods, in accordance with Oikawa et al. [6], since we use the future of R_{eco} predictors in addition. We are thereby able to potentially model the daytime R_{eco} forecasts more realistically with respect to these predictors. The study performed by Lee et al. [9] uses continuous stable isotope measurements in a Pacific Northwest Douglas-fir forest ecosystem for NEE partitioning and also reports estimated daytime R_{eco} lower than the conventional approaches. Furthermore, similar results were obtained in the most recent study on the topic by Kira et al. [10], in which the authors used a parsimonious Solar-Induced Chlorophyll Fluorescence (SIF)-based approach for NEE partitioning and examined its performance using synthetic simulations and field measurements.

Time series forecasting using deep learning. State space models have been used for decades as one of the main methods for sequence modeling [11]. Other related algorithms from a graphical model perspective are Kalman filters [12] and conditional Markov processes [13]. With the development of deep learning and variational inference resulting in the rise of Variational Auto Encoders (VAE) [14], several methods for sequential data based on VAEs have appeared as well. Chung et al. [15] propose a recurrent latent variable model for sequential data, namely the Variational Recurrent Neural Network. It can model variability such as that observed in highly structured data like natural speech through the use of high-level latent random variables. When it comes to other deep state space models for sequential data, Doerr et al. [16] propose a probabilistic recurrent state space model with a scalable training algorithm based on a doubly stochastic variational inference and Gaussian processes. Krishnan et al. [17] introduce the deep Kalman filter – a deep state space model, which can generate counterfactual data given an action variable, apart from its capability of time series forecasting. In 2017, Salinas et al. [18] introduced a probabilistic forecasting method based on the training of an autoregressive recurrent network model on a large number of related time series called DeepAR. A year later, two deep state space models were proposed. Namely, Zheng et al. [19] proposed the State Space LSTM models alongside a sampler based on sequential Monte Carlo method [20] that directly draws samples from the joint posterior. The other deep state space method, DeepState [2], parameterizes a per-time-series linear state space model with a jointly-learned RNN and scales from regimes with little training data to those where large amounts of time series are available to learn accurate models. It compares favorably to the state-of-the-art results on probabilistic time series forecasting tasks which is a reason why we employ it for the task of partitioning NEE.

B. OUTLINE OF THE PAPER

The ecological data is abundant and NEE has been measured by more than 200 FLUXNET towers all over the world for decades, yet there are very few machine learning approaches for NEE partitioning. The existing ones rely on approximations of the daytime $\mathrm{R_{eco}}$ using its nighttime values and creating proxies of GPP with the use of $\mathrm{NEE_{day}}$ which can incur additional approximation error. The goal of this paper is to introduce a purely data-driven approach to partition NEE based on a deep state space model DeepState [2], which we introduce in Section II. Instead of relying on other method's approximations of the daytime $\mathrm{R_{eco}}$, we use the known $\mathrm{R_{eco}}$ predictors' both the past and the future to obtain daytime $\mathrm{R_{eco}}$ forecasts having plausible relationships with those predictors. Our methodology is described in detail in Section III. Our method's performance compared to that of other time series forecasting methods and other NEE partitioning methods with respect to the ground truth $\mathrm{NEE_{night}}$ is presented in Section III-C. Promptly thereafter, in Section III-D, we present our daytime $\mathrm{R_{eco}}$ forecasts. Section IV concludes the paper.

II. DEEPSTATE

Before introducing the DeepState model [2] in more detail, we will first discuss linear state space models, as they are an important component of DeepState's architecture. General linear Gaussian state space models can be defined as follows:

$$z_t = A_t l_t + \eta_t, \quad \eta_t \sim \mathcal{N}(0,1) \tag{2}$$

$$l_t = F_t l_{t-1} + g_t \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1). \tag{3}$$

In Eqs. (2) and (3), z_t denotes a vector of observations, aptly called the *observation vector*, whereas l_t is an unobserved vector called the *state vector* or the *latent state*, both at time t, for $t = 1, \dots, T$, as per Durbin et al. [21]. Furthermore, by η_t, ϵ_t we denote the mutually-independent error terms and by A_t, F_t, g_t different matrices. Matrix A_t is called the *system matrix*. Latent state $l_t \in \mathbb{R}^L$ at time t, for $L \in \mathbb{N}$, changes according to the influence of the *transition matrix* F_t and a random *innovation* $g_t\epsilon_t$. The structure of F_t and the *innovation strength* g_t determine which type of time series patterns are encoded by the latent state l_t.

The desired benefits of using a state space model are estimating the underlying evolution of the unobserved signal l_t given the data z_i for $i = 1, \dots, s$, $s \in \mathbb{N}$ and the corresponding system parameter updates. If $s < t$, the problem is called *forecasting*. When $s = t$, it is called *filtering*, and if $s > t$, *smoothing*. We will concentrate on forecasting since we would like to learn a data-driven time-dependent model of $\mathrm{R_{eco}}$ and estimate its daytime values to partition NEE.

The deep state space method we are using is based on probabilistic forecasting. A probabilistic forecast consists of a probability density function which estimates the respective probability distributions for all possible future outcomes of a random variable. To describe the concept more formally, following Krishnan et al. [2], we let $\{z^{(i)}_{1:T_i}\}_{i=1}^N$ be a set of univariate *target* time series, where $z^{(i)}_{1:T_i} = (z^{(i)}_1, \dots, z^{(i)}_{T_i})$ and $z^{(i)}_t \in \mathbb{R}$ is the value of the i-th time series at time t, for $t = 1, \dots, T_i$ and $i = 1, \dots, N$, where $T_i, N \in \mathbb{N}$. Further, let $\{x^{(i)}_{1:T_i+\tau}\}_{i=1}^N$ be a set of associated covariate vectors with $x^{(i)}_t \in \mathbb{R}^D$, for $D, \tau \in \mathbb{N}$ and $i = 1, \dots, N$. The goal is to produce a set of probabilistic forecasts, i.e. for each $i = 1, \dots, N$, we are interested in the probability distribution of future trajectories $z^{(i)}_{T_i+1:T_i+\tau}$ given the past ones, as well as all the past and τ future samples of the covariate vectors:

$$p(z^{(i)}_{T_i+1:T_i+\tau} \mid z^{(i)}_{1:T_i}, x^{(i)}_{1:T_i+\tau}; \Phi). \tag{4}$$

In Eq. (4), Φ denotes the set of learnable parameters of the model, which are, in the case of DeepState, shared between and learned jointly from all N time series. For any $i = 1, \dots, N$, we refer to the time range $\{1, \dots, T_i\}$ as the *training range*, and to $\{T_i + 1, \dots, T_i + \tau\}$ as *prediction range*. We assume that the covariate vectors $x^{(i)}_t$ are given in both the training and the prediction range, as seen in Eq. (4).

Constructed as a fusion of deep learning and state space models, DeepState is a forecasting method that parameterizes a particular linear state space model (Eqs. (2) and (3)) using an RNN. The observations are generated from the latent state l_t according to the following model:

$$\begin{aligned} z_t &= y_t + \sigma_t \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1) \\ y_t &= a_t^\intercal l_{t-1} + b_t \end{aligned} \tag{5}$$

where $a_t \in \mathbb{R}^L$, $\sigma_t \in \mathbb{R}_{>0}$ and $b_t \in \mathbb{R}$ are additional model parameters. The initial state l_0 is assumed to follow an isotropic Gaussian distribution, namely $l_0 \sim \mathcal{N}(\mu_0, \mathrm{diag}(\sigma_0^2))$. The state space model is entirely determined by the parameters $\Theta_t = (\mu_0, \Sigma_0, F_t, g_t, a_t, b_t, \sigma_t)$, $\forall t > 0$. One way of estimating them is by solving the optimization problem:

$$\underset{\Theta_{1:t}}{\operatorname{argmax}}\; p_{SS}(z_{1:T} | \Theta_{1:T}), \tag{6}$$

where

$$\begin{aligned} p_{SS}(z_{1:T}|\Theta_{1:T}) &:= p(z_1|\Theta_1) \prod_{t=2}^{T} p(z_t|z_{1:t-1}, \Theta_{1:t}) \\ &= \int p(l_0) \left[\prod_{t=1}^{T} p(z_t|l_t) p(l_t|l_{t-1}) \right] dl_{0:T} \end{aligned} \tag{7}$$

denotes the marginal probability of the observations $z_{1:T}$, given the parameters Θ under the state space model.
DeepState learns a function

$$\Psi(x_{1:t}^{(i)}, \Phi) = \Theta_t^{(i)}, \quad i = 1, \dots, N, \ t = 1, \dots, T_i + \tau \quad (8)$$

which maps the covariate vectors $x_{1:T_i}^{(i)}$ associated with each target time series $z_{1:T_i}^{(i)}$, as well as a set of shared parameters Φ, to the (time-dependent) parameters $\Theta_t^{(i)}$ of a linear state space model for the i-th time series. This is done globally, over all time series, instead of learning the state space parameters for each time series independently. This mapping is parameterized by a multi-layer RNN with LSTM cells [22] and parameters from Φ. The parameters of the RNN are learned jointly from a dataset of raw time series and their covariates, so that the model can extract features and learn complex temporal patterns. During training, the inputs to the network are the covariate vectors $x_t^{(i)}$, together with the previous network output $h_{t-1}^{(i)}$ at each time step t in the training range $\{1, \dots, T_i\}$. Then, the network output, computed by a recurrent function h, namely $h_t^{(i)} = h(h_{t-1}^{(i)}, x_t^{(i)}, \Phi)$ is used to determine the parameters $\Theta_t^{(i)}$ of the state space model. Those parameters are then used to obtain the likelihood of the given observations $z_{1:T_i}^{(i)}$, as seen in Eq. (7). The shared network parameters Φ are then learned by maximizing the likelihood $\mathcal{L}(\Phi)$ given by Eq. (9).

$$\begin{aligned} \mathcal{L}(\Phi) &= \sum_{i=1}^{N} \log p(z_{1:T_i}^{(i)} \mid x_{i:T_i}^{(i)}, \Phi) \\ &= \sum_{i=1}^{N} \log p_{SS}(z_{1:T_i}^{(i)} \mid \Theta_{i:T_i}^{(i)}). \end{aligned} \quad (9)$$

The forecasts are obtained as follows, given a target time series $z_{1:T_i}^{(i)}$ in the training range i.e. for time steps $1, \dots, T_i$, and associated covariate vectors $x_{1:T_i+\tau}^{(i)}$ in both the training and the prediction range. Namely, first the posterior of the latent state $p(l_{T_i}|z_{1:T_i})$ is computed for the last time step T_i in the training range using the observations $z_{1:T_i}^{(i)}$ and the state space parameters $\Theta_{1:T_i}^{(i)}$ obtained by unrolling the RNN network in the training range. Given the posterior of the latent state $p(l_{T_i}|z_{1:T_i})$, prediction samples are generated by recursively applying the transition equation and the observation model where the state space parameters for the prediction range $\Theta_{T_i+1:T_i+\tau}^{(i)}$ are obtained by unrolling the RNN in the prediction range. An illustration of the DeepState forecast with $\text{NEE}_{\text{night}}$ as the observation can be seen in Fig. 1. Lower and upper bounds for the diagonal of the prior covariance matrix Σ of the latent state l_t are set to 10^{-6} and 1 by default, respectively. Lower and upper bounds for the standard deviation of the innovation function g_t are by default set to 10^{-6} and 0.01, respectively. Throughout this paper, we refer to covariate vectors as *predictor variables* or *predictors*.

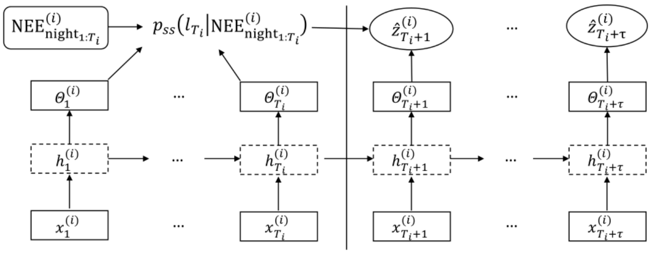


FIGURE 1: **Illustration of forecasting using DeepState.** Here we substitute observations z by $\text{NEE}_{\text{night}}$ to portray our training data. By $\hat{z}^{(i)}$ we denote the forecasted values of R_{eco} for a given sample $i \in \{1, \dots, N\}$, with $T_i = 288$ and $\tau = 48$. In our setup, vector $x^{(i)}$ consists of a subset of R_{eco} predictors R_g, T_{air}, PPT, VPD and NEE_{day} in both training and prediction ranges separated by the vertical line.

III. PARTITIONING NEE USING DEEPSTATE

Our NEE partitioning approach relies on DeepState [2] to learn the dynamical model of nighttime R_{eco} and estimate it during the day. This estimation is aided not only by the past (days and nights) of the predictor variables but also by their values during the period in which we want to estimate R_{eco}, i.e. in the prediction range. Once the daytime R_{eco} forecast is obtained, it is possible to subtract it from the observed NEE_{day} values to obtain GPP and thus partition NEE.

A. METHODOLOGY

We make use of the fact that $\text{NEE}_{\text{night}}$ equals the nighttime R_{eco}, thus putting our approach in the category of the nighttime partitioning methods [23]. However, in contrast to $\text{NN}_{\text{C-part}}$ method [1], for instance, we do not average the nighttime R_{eco} values to leverage the lack of daytime R_{eco}, since DeepState can handle missing values during training. More precisely, DeepState approximates the distribution of the target variable and samples from it during forecasting, so there can be no missing values in the forecasts. We separate the NEE observations during day and night by retrieving exact times of sunrise and sunset for each site according to its timezone. We increase these values by one hour to ensure that there is enough daylight to activate photosynthesis in the morning and that there is no daylight in the evening in order to separate nighttime observations. We train on the $\text{NEE}_{\text{night}}$ observations, which we do not normalize, to learn the dynamical model of R_{eco} as a function of its previous values. To achieve this even during the daytime, we use different R_{eco} predictors R_g, T_{air}, VPD and PPT, as well as NEE_{day} as covariate feature vectors of the DeepState model. In regard to the model parameters, we use the batch size of 32, the dropout rate of 0.1, the learning rate of 10^{-4}, and the LSTM configuration of two layers and 40 cells. The training samples consist of half-hourly $\text{NEE}_{\text{night}}$ observations spanning one week, originating from the four summer months, by shifting the aforementioned one-week window by two days. We train on six days and use the last day for testing. Training on longer samples, i.e. 13 or 55 days

	FR-Pue	DE-Hai	IT-Ro1	NL-Loo	BR-Sa3	CG-Tch
Vegetation type	EBF	DBF	DBF	ENF	EBF	SAV
T_{air} (°C)	13.5	8.3	15.15	9.8	26.12	25.7

TABLE 1: **Vegetation type and annual mean temperature of different FLUXNET sites.** Evergreen Broadleaf Forests are referred to as EBF, Deciduous Broadleaf Forests as DBF, Evergreen Needleleaf Forests as ENF, and Savannas as SAV.

did not improve the results. We showcase the DeepState's training and test loss over 600 epochs in Fig. 5 given in the Appendix A. Adapting the hyperparameters of the DeepState such as the number of training periods when using different predictor variables did not yield a considerable difference in the forecasts. Furthermore, we tested different number of LSTM layers, dropout rates and learning rates while predicting the NEE_{night} as shown in Fig. 6 given in the Appendix A. This type of hyperparameter tuning also did not substantially improve the DeepState's NEE_{night} forecasts. Moreover, the hyperparameter configuration which produces this slight improvement differs for each FLUXNET site and each subset of the predictor variables. Therefore, in order to fix a hyperparameter configuration that achieves the best NEE_{night} forecasts on average, we chose the above-mention experimental setup to facilitate the application of our NEE partitioning method.

We forecast one full day of R_{eco} which can then be subtracted from NEE for obtaining GPP. The quantitative evaluation is performed for the nighttime samples to assess the prediction accuracy of the learned night R_{eco} model with respect to the ground truth NEE_{night}. Due to the non-stationarity of the training data, our method sometimes yields suboptimal nighttime R_{eco} forecasts, however this occurs in less than 3% of the forecasts and are shown in Fig. 7 given in the Appendix B. The interpretation of the daytime R_{eco} forecast obtained by using its learned model while conditioning on different daytime values of the predictors will be discussed in Section III-D with respect to the multiple empirical studies as there is no ground truth daytime R_{eco}.

We apply our NEE partitioning approach to data from the FLUXNET sites of different climate regions. European sites in France (FR-Pue), Germany (DE-Hai), the Netherlands (NL-Loo), and Italy (IT-Ro1), as well as equatorial sites in Brazil (BR-Sa3) and Congo (CG-Tch) were analyzed. Table 1 shows an overview of vegetation types and the average annual temperature at each site. The sites are chosen to enable a fair overview in context of other NEE partitioning methods and provide a broader analysis of our NEE partitioning method across different vegetation and climate conditions. We selected the FLUXNET sites IT-Ro1 and NL-Loo specifically so that the correlation of T_{air} and R_{eco} is the lowest compared to all the sites we use. We did this to prevent the DeepState from simply learning the T_{air} dynamics and producing good R_{eco} forecasts due to the high correlation between these two variables. The data is selected from June to September of each year in the case of sites FR-Pue, DE-Hai,

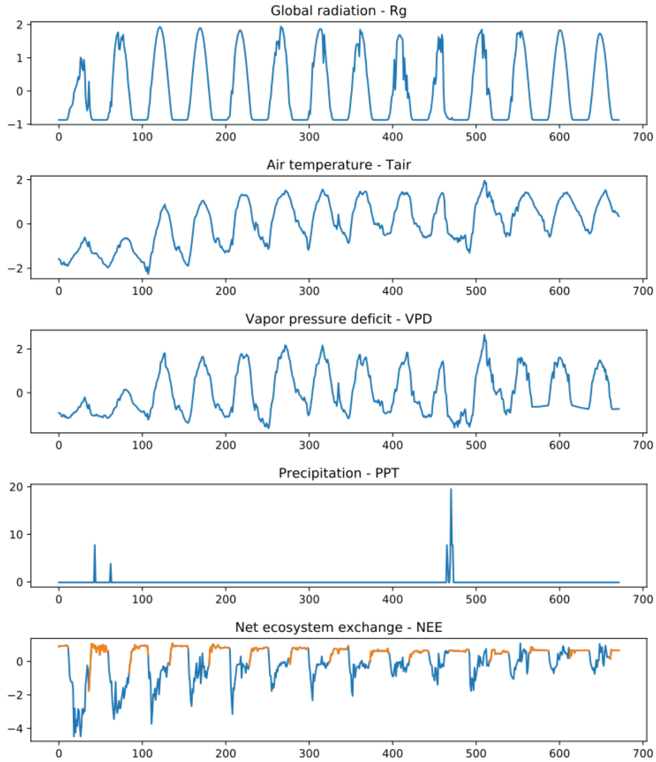


FIGURE 2: **Two-week data sample - FR-Pue.** Two weeks of half-hourly R_g, T_{air}, VPD, PPT and NEE, measured at the French site Puechabon. Nighttime R_{eco} used for training DeepState is shown in orange.

IT-Ro1, and NL-Loo. For BR-Sa3 and CG-Tch, we selected data from January to April of each year. These equatorial sites are chosen to analyze how different climate regions and different T_{air} and PPT variation during the day vs during the night influence the learned R_{eco} model. As R_{eco} predictors, we use SW_IN_F_MDS, TA_F_MDS, P_F, VPD_F_MDS and NEE_VUT_USTAR50 during the daytime, corresponding to R_g, T_{air}, PPT, VPD, and NEE_{day}, respectively.

B. EVALUATION METRICS

The error metrics we use for evaluating our method in comparison with the other partitioning methods with respect to the ground truth NEE_{night} are the Root Mean Square Error (RMSE) and the Mean Absolute Percentage Error (MAPE), defined as:

$$\text{RMSE}(P, R) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(R_t - P_t)^2} \tag{10}$$

$$\text{MAPE}(P, R) = \frac{1}{T}\sum_{t=1}^{T}\frac{|R_t - P_t|}{|R_t|} \tag{11}$$

In Eq. (10) and Eq. (11), by $R = [R_1, \ldots, R_T]$ we denote the vector of real target values and by $P = [P_1, \ldots, P_T]$ the vector of prediction values, while $T \in \mathbb{N}$ denotes the number of time steps.

Predictors					FR-Pue MAPE	FR-Pue RMSE	DE-Hai MAPE	DE-Hai RMSE	BR-Sa3 MAPE	BR-Sa3 RMSE	CG-Tch MAPE	CG-Tch RMSE
		None			0.22	0.78	0.18	0.82	1.02	3.95	0.21	0.54
R_g					0.21	0.76	0.17	0.78	0.9	3.93	0.19	0.48
	T_{air}				0.21	0.76	0.17	0.79	0.96	3.91	0.21	0.5
		PPT			0.23	0.79	0.19	0.81	1.09	4.07	0.2	0.52
			VPD		0.21	0.77	0.18	0.8	0.92	3.93	0.21	0.5
				NEE_{day}	0.21	0.73	0.18	0.8	1.45	4.3	0.24	0.49
R_g	T_{air}				0.2	0.73	0.16	0.74	0.96	3.93	0.18	0.45
R_g		PPT			0.22	0.77	0.17	0.78	0.91	3.92	0.19	0.48
R_g			VPD		0.2	0.73	0.17	0.77	0.95	3.92	0.21	0.5
R_g				NEE_{day}	0.2	0.73	0.17	0.77	0.93	3.9	0.19	0.48
	T_{air}	PPT			0.21	0.76	0.17	0.78	0.95	3.92	0.2	0.49
	T_{air}		VPD		0.2	0.73	0.18	0.79	0.96	3.92	0.21	0.5
	T_{air}			NEE_{day}	0.2	0.73	0.17	0.77	0.96	3.93	0.2	0.48
		PPT	VPD		0.21	0.75	0.18	0.79	0.96	3.94	0.2	0.48
		PPT		NEE_{day}	0.2	0.73	0.18	0.8	**0.69**	**3.87**	0.21	0.5
			VPD	NEE_{day}	0.2	0.74	0.17	0.77	0.96	3.93	0.2	0.49
R_g	T_{air}	PPT			0.2	0.74	0.16	0.75	0.95	3.9	**0.18**	**0.44**
R_g	T_{air}		VPD		0.19	0.71	0.16	0.74	0.95	3.91	0.18	0.45
R_g	T_{air}			NEE_{day}	0.19	0.69	0.16	0.76	0.95	3.89	0.18	0.46
R_g		PPT	VPD		0.2	0.73	0.16	0.75	0.93	3.88	0.19	0.45
R_g		PPT		NEE_{day}	0.2	0.71	0.17	0.79	0.86	3.88	0.19	0.45
R_g			VPD	NEE_{day}	0.18	0.69	0.17	0.77	0.93	3.88	0.19	0.48
	T_{air}	PPT	VPD		0.2	0.73	0.17	0.78	0.95	3.9	0.2	0.48
	T_{air}	PPT		NEE_{day}	0.2	0.72	0.17	0.79	0.87	3.93	0.2	0.48
	T_{air}		VPD	NEE_{day}	0.19	0.7	0.17	0.77	0.95	3.91	0.2	0.48
		PPT	VPD	NEE_{day}	0.2	0.74	0.18	0.79	0.95	3.89	0.2	0.48
R_g	T_{air}	PPT	VPD		0.19	0.71	0.16	0.75	0.95	3.89	0.18	0.45
R_g	T_{air}	PPT		NEE_{day}	0.19	0.7	0.16	0.75	0.94	3.9	0.18	0.45
R_g	T_{air}		VPD	NEE_{day}	**0.17**	**0.66**	**0.16**	**0.73**	0.95	3.89	0.18	0.45
R_g		PPT	VPD	NEE_{day}	0.19	0.71	0.17	0.77	0.94	3.88	**0.18**	**0.44**
	T_{air}	PPT	VPD	NEE_{day}	0.2	0.72	0.17	0.78	0.95	3.91	0.2	0.47
R_g	T_{air}	PPT	VPD	NEE_{day}	0.18	0.66	0.16	0.74	0.95	3.87	0.18	0.45

TABLE 2: **NEE_{night} estimation accuracy.** Quantitative results of forecasting one day of nighttime R_{eco}, using different predictor variables for training DeepState on sites FR-Pue, DE-Hai, BR-Sa3 and CG-Tch. The best results are shown in bold.

C. LEARNING A DATA-DRIVEN TIME-DEPENDENT MODEL OF NIGHTTIME R_{ECO}

For training DeepState, we used the past values of NEE_{night}, i.e. the nighttime R_{eco}, together with the past and the future of R_g, T_{air}, VPD, PPT and NEE_{day} in all possible combinations as predictors. This allowed the model to learn the underlying dynamics of R_{eco} at night on multiple FLUXNET sites. We evaluate how well this nighttime dynamical model is learned by comparing our nighttime R_{eco} forecasts with the ground truth NEE_{night} on multiple sites, as shown in Table 2. We find that using R_g, T_{air}, VPD and NEE_{day} as predictors provides the best ground-truth NEE_{night} estimates on the European sites FR-Pue and DE-Hai. For the equatorial site BR-Sa3, the best predictors are PPT and NEE_{day}, whereas the combination of predictors R_g, T_{air} and PPT, and the combination of R_g, PPT, VPD and NEE_{day} yield the best NEE_{night} estimates on the Congolese site CG-Tch.

We compared the DeepState's nighttime R_{eco} forecasts to the corresponding R_{eco} estimates of other partitioning methods and to two other time series forecasting methods in order to justify our choice of this deep state space model. Namely, we compared to the Reichstein method, the $NN_{C\text{-part}}$, as well as to a more traditional time series forecasting method ARIMA [24] and to the RNN-based deep learning method for time series forecasting, namely, DeepAR [18]. In the case of ARIMA however, since our training data contains only the nighttime R_{eco} values, the missing daytime values are imputed by a Kalman filter [12]. In Table 3 we present the quantitative comparison of the nighttime R_{eco} to the ground truth NEE_{night} for all the sites we apply our method to when using different partitioning and time series forecasting methods. We conclude that DeepState provides the best nighttime R_{eco} forecasts with respect to the ground truth NEE_{night} in

Method	FR-Pue	DE-Hai	NL-Loo	IT-Ro1	CG-Tch	BR-Sa3
DeepState	0.17	0.16	**0.51 (0.69)**	**0.11 (0.3)**	**0.18**	**0.69**
$NN_{C\text{-part}}$	/	0.24	0.54	0.18	/	/
Reichstein	0.26	0.24 (0.25)	0.56 (0.9)	0.14 (0.58)	0.54	1.4
ARIMA	**0.16**	**0.15**	0.52	0.31	0.4	1.38
DeepAR	0.24	0.2	0.73 (0.75)	0.45 (0.42)	0.4	1.41

TABLE 3: **Comparison of different methods' nighttime R_{eco} forecasts to the ground truth NEE_{night}.** For the sites where $NN_{C\text{-part}}$ can be applied, we show the NEE_{night} estimation MAPE error metric of other methods both on the same (reduced) amount of data samples as provided by the $NN_{C\text{-part}}$ method, and on the entire data in brackets when these two values differ. The best results on each site are shown in bold.

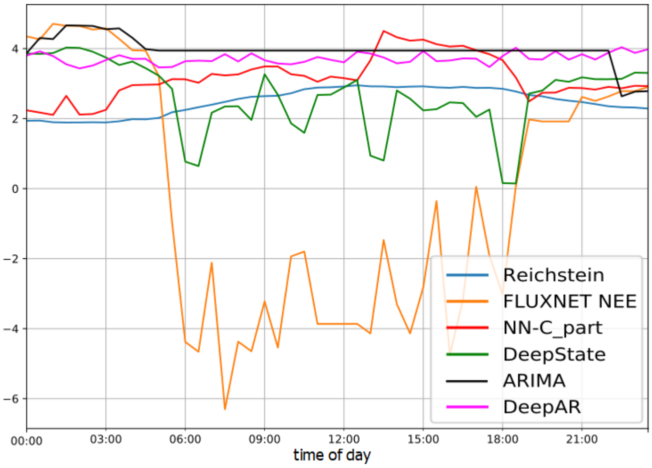


FIGURE 3: **Nighttime R_{eco} qualitative evaluation.** One day R_{eco} forecast of different time series forecasting methods and NEE partitioning methods on the site IT-Ro1.

comparison to other partitioning methods, i.e. Reichstein and $NN_{C\text{-}part}$, and attains very similar MAPE values to those of ARIMA on the two European sites FR-Pue and DE-Hai. Even though ARIMA sometimes performs slightly better, it is not suitable for the daytime R_{eco} forecasts, as depicted in Fig. 3 for the site IT-Ro1. Moreover, the ARIMA model tends to be difficult to fit due to the non-stationarity of the data.

We consider the R_{eco} forecasts from midnight to 5 a.m. and from the 7 p.m. until 11:59 p.m. to denote the nighttime. We note that in the period from midnight to 5 a.m. our method produces considerably better nighttime R_{eco} forecasts than the other NEE partitioning approaches. Furthermore, our R_{eco} forecasts using T_{air} as a predictor are depicted in Fig. 4 for each of the six above-mentioned sites. The DeepState's ability to model the ground truth NEE_{night} better than the Reichstein's method is particularly visible in Fig. 4e and Fig. 4f on the sites BR-Sa3 and CG-Tch, respectively. This is expected since we use the past NEE_{night} values during the training phase, but the results on other sites in Fig. 4 show that DeepState fits the nighttime NEE_{night} better than $NN_{C\text{-}part}$ as well. Additionally, the identification of the best predictors gives us a valuable insight into which variables to use when partitioning NEE. In addition, the results from Table 2 confirm that T_{air} is a good R_{eco} predictor, as suggested by Platenius in 1942 [25]. Namely, when considering a single R_{eco} predictor, the NEE_{night} estimation error metrics when using T_{air} tend to be among the two best ones across all sites.

D. DAYTIME R_{ECO} FORECAST EVALUATION

In the previous section we have seen how well DeepState learns the dynamical model of R_{eco} at night, as well as how it is trained. Now we can interpret our daytime R_{eco} forecasts having in mind that there is no ground truth with which to compare these forecasts directly. We compare our findings to those of the recent empirical NEE partitioning methods [6], [9], [10] which suggest that R_{eco} is lower during the day than during the night, and can be attributed to the Kok effect [8]. In Fig. 4, we observe that this is also the case with our daytime R_{eco} forecasts on most of the inspected sites. Oikawa et al. [6] report on average 10-13% lower daytime R_{eco} than that of the Reichstein method, whereas our forecasts are on average 16-17% lower than the Reichstein method's R_{eco}. The lower R_{eco} forecasts during the day might also be attributed to the findings of Wehr et al. [26], i.e. that the nighttime R_{eco} is twice as large as daytime R_{eco} during the first half of the growing season in a temperate forest. The $NN_{C\text{-}part}$ method's R_{eco} estimates also tend to be slightly lower than the Reichstein method's, but are very sparse due to many missing values in the training data. The influence of the missing data on the $NN_{C\text{-}part}$ model's performance is reflected in the fact that there are much less output R_{eco} estimation values and that some FLUXNET sites are not considered for partitioning using $NN_{C\text{-}part}$ due to many missing values in any given year for which a threshold can be set. In contrast, since DeepState's predictions are sampled from the estimated probability distribution, the missing values do not occur in our forecasts. Moreover, deep state space models are more robust to the missing values during training which makes our method applicable to any FLUXNET site without further restrictions.

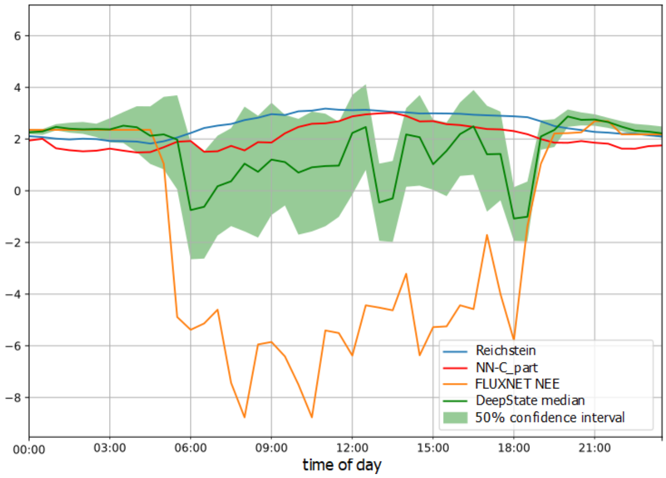
On certain sites, such as CG-Tch, depicted in Fig. 4f, our daytime R_{eco} forecasting is fluctuating more than expected in comparison to the other two partitioning methods, but as shown empirically by Oikawa et al. [6], this behaviour can sometimes indeed occur.

IV. CONCLUSION

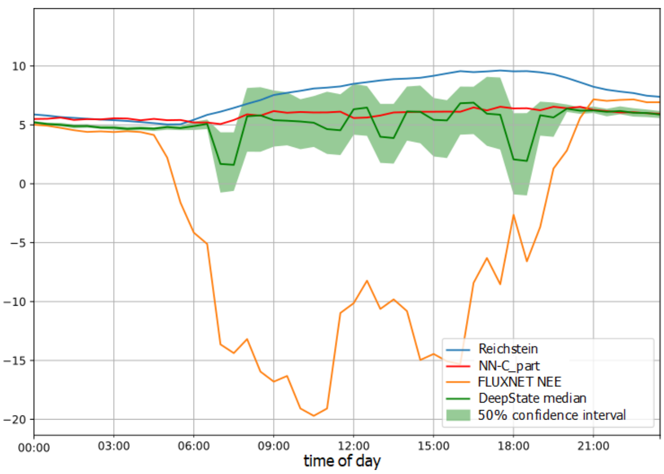
In this paper we have proposed an NEE partitioning approach based on the deep probabilistic state space model DeepState, thus leveraging the interpretability and uncertainty analysis of state space models with the advantages of deep networks in learning complex dynamical models. The quantitative analysis of our method for estimating R_{eco} during the nighttime justified our choice of DeepState for learning the dynamical model of R_{eco}. Furthermore, our daytime R_{eco} forecasts, aided by both the past and the future of the R_{eco} predictors, are in line with the empirical studies which suggest that R_{eco} during the day is usually lower than that of the Reichstein method. This discrepancy can also be attributed to the observation that nighttime R_{eco} is twice as high as the daytime R_{eco} in temperate forests during the first half of the growing season [26] and to the Kok effect [8]. Therefore, we conclude that our approach offers a good alternative to data-driven NEE partitioning across a diverse set of vegetation types when the right R_{eco} predictors are selected and complements other NEE partitioning methods.

REFERENCES

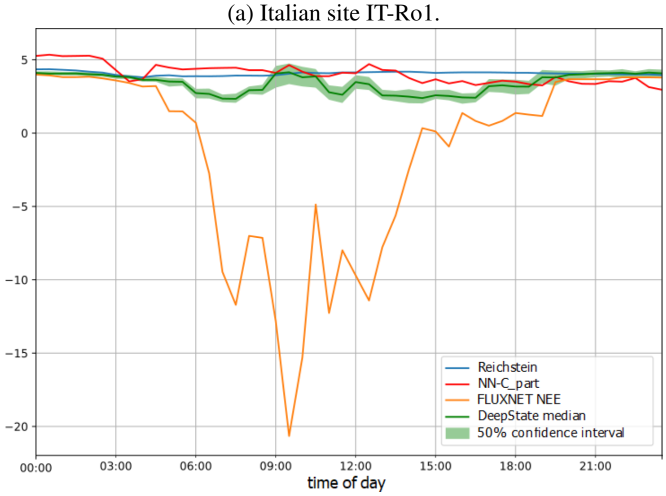
[1] G. Tramontana, M. Migliavacca, M. Jung, et al., “Partitioning net carbon dioxide fluxes into photosynthesis and respiration using neural networks,” Global Change Biology, vol. 26, no. 9, pp. 52355–5253, 2020.



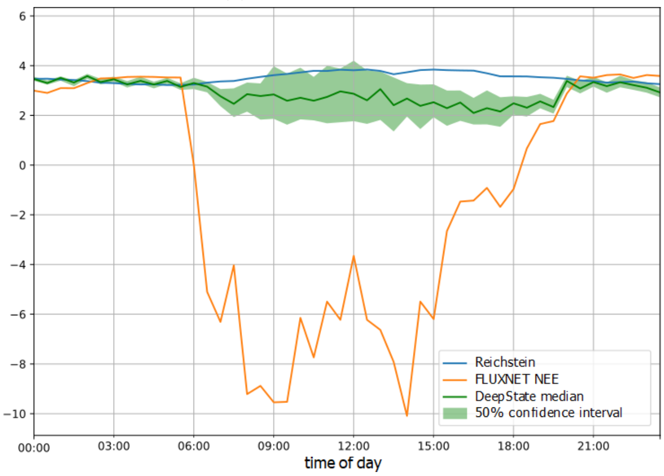
(a) Italian site IT-Ro1.



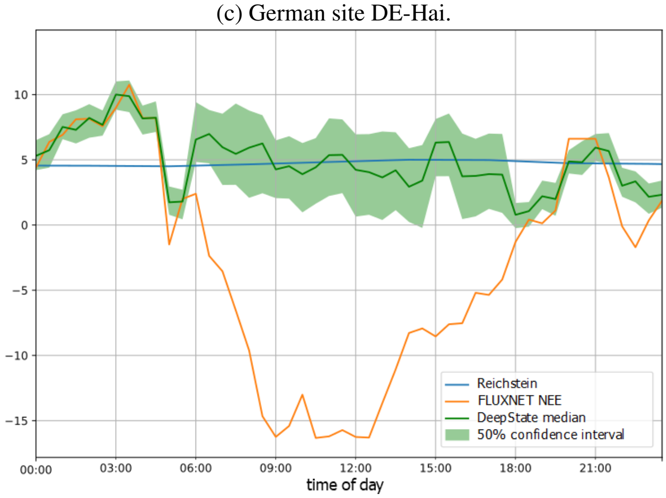
(b) Dutch site NL-Loo.



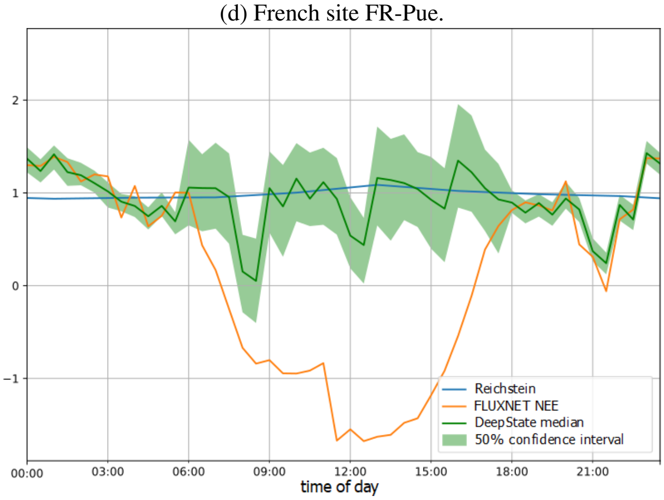
(c) German site DE-Hai.



(d) French site FR-Pue.



(e) Brazilian site BR-Sa3.



(f) Congolese site CG-Tch.

FIGURE 4: **One-day R_{eco} forecast (green) using DeepState with T_{air} as a predictor.** Observed NEE is shown in orange, the R_{eco} obtained by the Reichstein method in blue and R_{eco} estimated by the $NN_{C\text{-}part}$ method is shown in red.

[2] S. S. Rangapuram, M. W. Seeger, J. Gasthaus, et al., "Deep state space models for time series forecasting," in Advances in Neural Information Processing Systems (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[3] M. Reichstein, E. Falge, D. Baldocchi, et al., "On the separation of net ecosystem exchange into assimilation and ecosystem respiration: review and improved algorithm," Global Change Biology, vol. 11, no. 9, pp. 1424–

1439, 2005.
[4] G. Lasslop, M. Reichstein, D. Papale, et al., "Separation of net ecosystem exchange into assimilation and respiration using a light response curve approach: critical issues and global evaluation," Global Change Biology, vol. 16, no. 1, pp. 187–208, 2010.
[5] I. J. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2019.
[6] P. Oikawa, C. Sturtevant, S. Knox, et al., "Revisiting the partitioning of net ecosystem exchange of co_2 into photosynthesis and respiration with simultaneous flux measurements of $^{13}co_2$ and co_2, soil respiration and a biophysical model, canveg," Agricultural and Forest Meteorology, vol. 234-235, pp. 149–163, 2017.
[7] A. M. Moffat, D. Papale, and M. Reichstein, "Comprehensive comparison of gap-filling techniques for eddy covariance net carbon fluxes," Agricultural and Forest Meteorology, vol. 147, pp. 209–232, 2007.
[8] B. Kok, "A critical consideration of the quantum yield of chlorella-photosynthesis," Enzymologia, vol. 13, pp. 1–56, 1948.
[9] S.-C. Lee, A. Christen, T. A. Black, et al., "Partitioning of net ecosystem exchange into photosynthesis and respiration using continuous stable isotope measurements in a pacific northwest douglas-fir forest ecosystem," Agricultural and Forest Meteorology, vol. 292-293, p. 108109, 2020.
[10] O. Kira, C. Y.-T. Chang, L. Gu, et al., "Partitioning net ecosystem exchange (nee) of co_2 using solar-induced chlorophyll fluorescence (sif)," Geophysical Research Letters, vol. 48, no. 4, p. e2020GL091247, 2021.
[11] L. Rabiner and B. Juang, "An introduction to hidden markov models," IEEE ASSP Magazine, vol. 3, no. 1, pp. 4–16, 1986.
[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," ASME Journal of Basic Engineering, vol. 82, pp. 35–45, 1960.
[13] R. L. Stratonovich, "Conditional markov processes," Theory of Probability and Its Applications, vol. 5, no. 2, pp. 156–178, 1960.
[14] D. Kingma and M. Welling, "Auto-encoding variational bayes," in Proceedings of International Conference for Learning Representations (ICLR), pp. 1—-14, 2014.
[15] J. Chung, K. Kastner, L. Dinh, et al., "A recurrent latent variable model for sequential data," in Advances in Neural Information Processing Systems (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., 2015.
[16] A. Doerr, C. Daniel, M. Schiegg, et al., "Probabilistic recurrent state-space models," in Proceedings of the 35th International Conference on Machine Learning (PMLR), vol. 80, pp. 1280–1289, 2018.
[17] R. G. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17, p. 2101–2109, AAAI Press, 2017.
[18] D. Salinas, V. Flunkert, J. Gasthaus, et al., "Deepar: Probabilistic forecasting with autoregressive recurrent networks," International Journal of Forecasting, vol. 36, no. 3, pp. 1181–1191, 2020.
[19] X. Zheng, M. Zaheer, A. Ahmed, et al., "State space lstm models with particle mcmc inference," in Conference on Neural Information Processing Systems (NIPS) workshop on Bayesian Deep Learning, 2017.
[20] A. Doucet, N. de Freitas, and N. Gordon, An Introduction to Sequential Monte Carlo Methods. Springer New York, 2001.
[21] J. Durbin and S. J. Koopman, Time Series Analysis by State Space Methods. Oxford University Press, 2001.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735—1780, 1997.
[23] M. Reichstein, P. C. Stoy, A. R. Desal, et al., "Partitioning of net fluxes," in Eddy Covariance: A Practical Guide to Measurement and Data Analysis (M. Aubinet, T. Vesala, and D. Papale, eds.), pp. 263–289, Springer Netherlands, 2012.
[24] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, Time series analysis: forecasting and control. Englewood Cliffs, N.J., Prentice Hall, 1994.
[25] H. Platenius, "Effect of temperature on the respiration rate and the respiratory quotient of some vegetables," Plant Physiology, vol. 17, no. 2, pp. 179–197, 1942.
[26] R. Wehr, J. W. Munger, and J. B. Mcmanus, "Seasonality of temperate forest photosynthesis and daytime respiration," Nature, vol. 534, no. 7609, pp. 680–683, 2016.

.

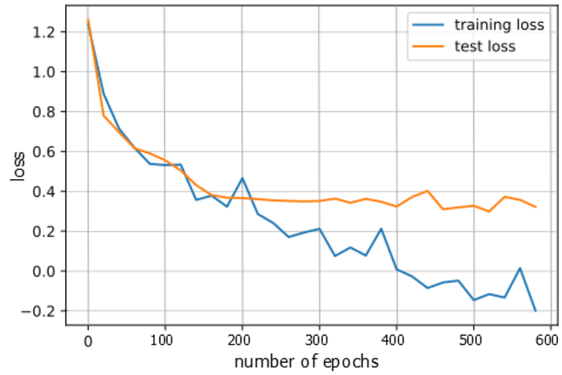


FIGURE 5: **DeepState's training and test loss.**

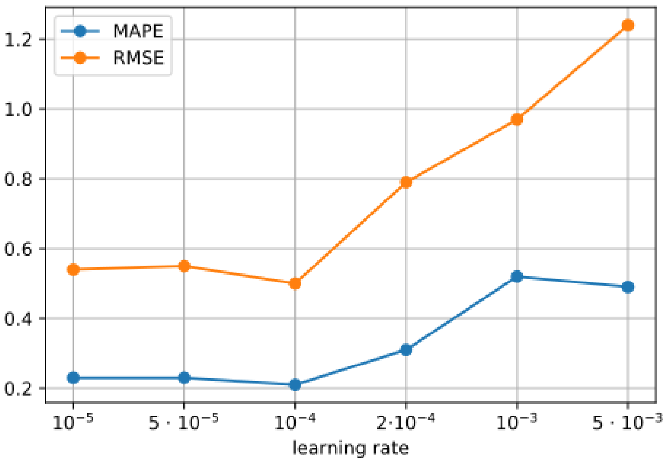
APPENDIX A DEEPSTATE TRAINING PROCESS

In Fig. 5 we depict the DeepState's training and test loss after each 20th epoch. The training is done for 600 epochs without validation, but we note that the forecasts are no longer improving after about 300 epochs when the plateau occurs. We can therefore stop after 600 epochs despite the difference between the training and the test loss becoming larger if trained further. The training data is from the Congolese site CG-Tch when using T_{air} as a predictor. We note that the loss can be negative as DeepState optimizes the negative likelihood of the Student-t distribution.

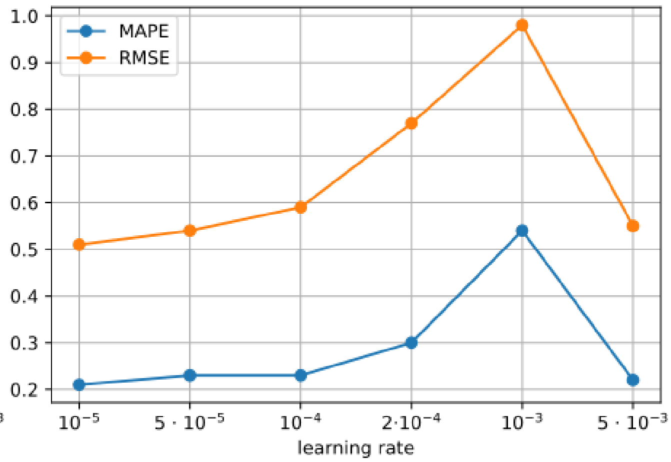
In Fig. 6, we show the MAPE and the RMSE metrics of the DeepState's nighttime R_{eco} forecast with respect to the ground truth NEE_{night}. The forecasts are performed on the data from the site CG-Tch with T_{air} as a predictor, as was previously the case for calculating the training and the test losses above. We varied the number of the LSTM layers between 2, 3 and 5, the dropout rate between 0.1 and 0.5, whereas the learning rates we tested are $10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 2 \cdot 10^{-4}, 10^{-3}$ and $5 \cdot 10^{-3}$. The configuration that we used throughout all our experiments, i.e., 2 LSTM layers, the dropout rate of 0.1 and the learning rate of 10^{-4} provides the MAPE of 0.21 and the RMSE metric of 0.5, whereas tuning the above-mentioned hyperparameters lowered these error metrics by 0.01 when using 3 LSTM layers. Using 2 layers, however, provided the best NEE_{night} forecasts on average, which is why we fixed this hyperparameter configuration for all our experiments and thus facilitated the use of our NEE partitioning approach.

APPENDIX B SUBOPTIMAL R_{ECO} FORECASTS

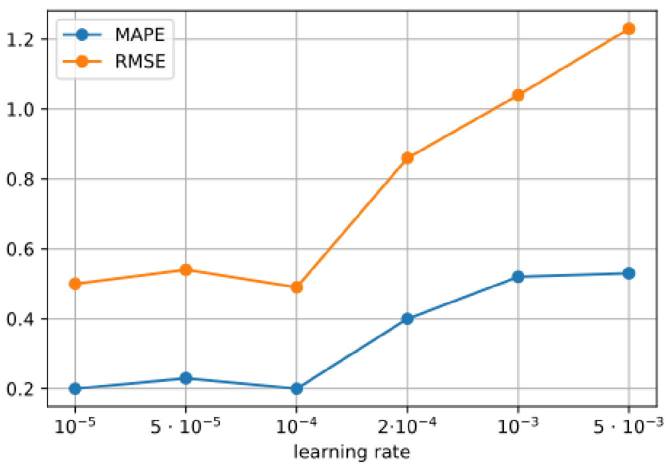
We here present our method's suboptimal forecasts. Namely, in Fig. 7, we illustrate samples where our method does worse at reproducing NEE_{night}, as well as in comparison to the FLUXNET R_{eco} baseline. This is mostly due to more drastic changes of the NEE levels of the prediction range compared with those during training. These forecasts, however, occur in less than 3% of all forecasts over all tested sites.



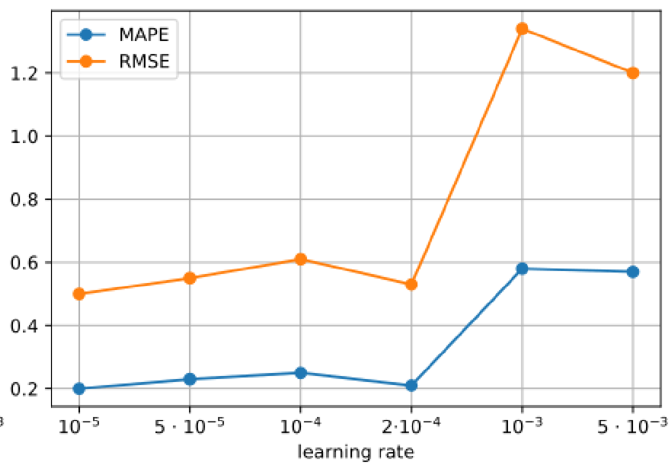
(a) 2 LSTM layers and dropout rate of 0.1



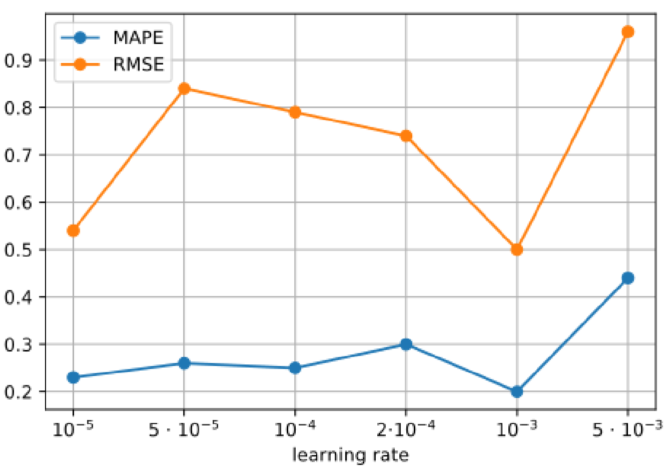
(b) 2 LSTM layers and dropout rate of 0.5



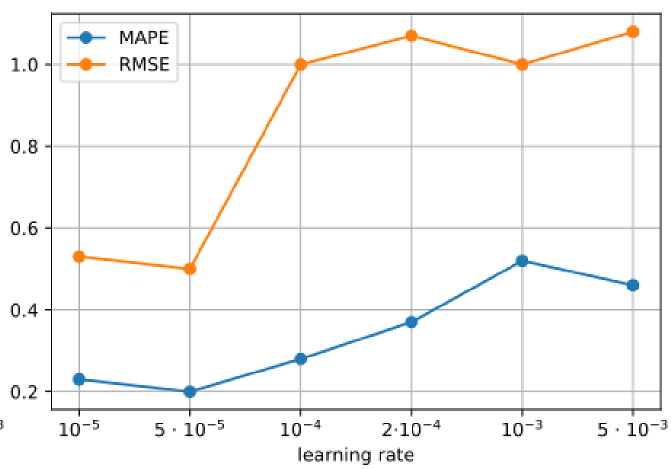
(c) 3 LSTM layers and dropout rate of 0.1



(d) 3 LSTM layers and dropout rate of 0.5

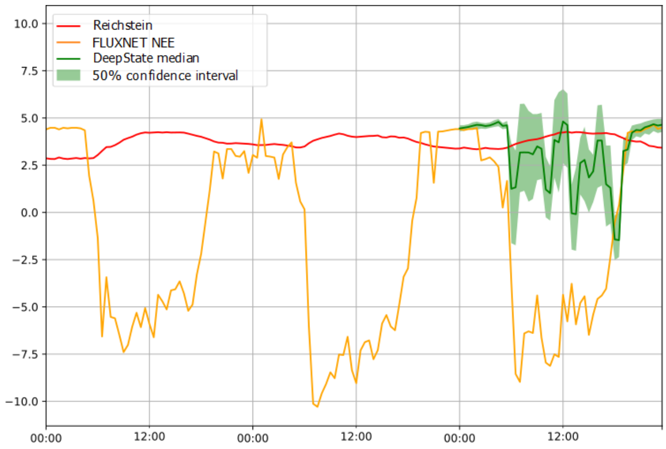


(e) 5 LSTM layers and dropout rate of 0.1

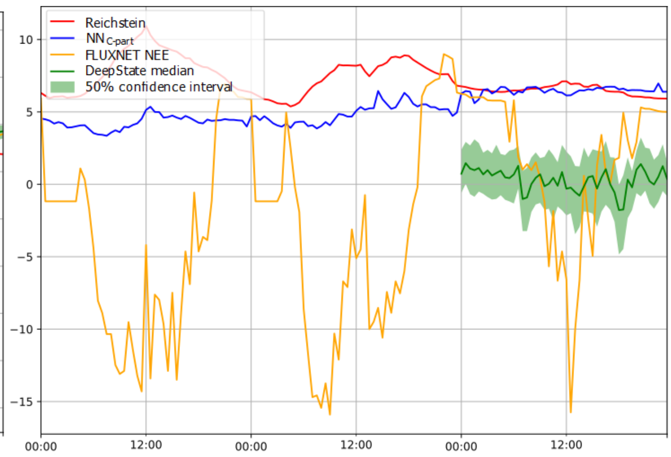


(f) 5 LSTM layers and dropout rate of 0.5

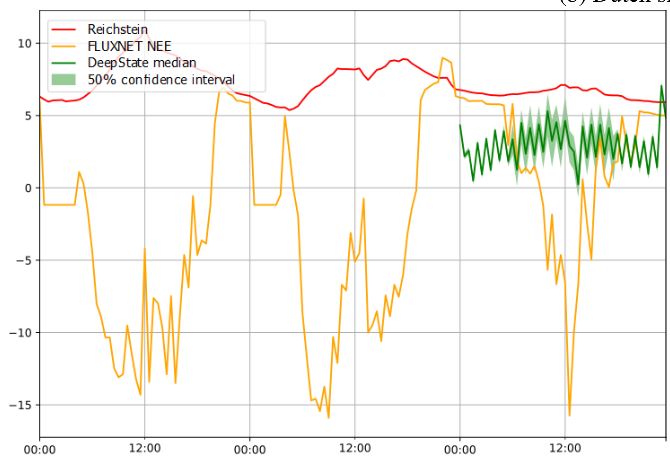
FIGURE 6: **Hyperparameter optimization.** Nighttime NEE forecast accuracy on the site CG-Tch using T_{air} as a predictor, for a different number of the LSTM layers and dropout rates, while varying the learning rate.



(a) Italian site IT-Ro1.



(b) Dutch site NL-Loo.

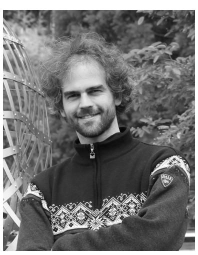


(c) Congolese site CG-Tch.

FIGURE 7: **Sub-optimal forecasts.** In green we show the one-day R_{eco} forecast from NEE_{night} on different FLUXNET sites with T_{air} as a predictor. Observed NEE is shown in orange, the FLUXNET R_{eco} in red and R_{eco} estimated by the $NN_{C\text{-part}}$ method is shown in blue.

VIOLETA TEODORA TRIFUNOV was born in Novi Sad, Serbia on the 7th of March 1993. She obtained the B.Sc. degree in theoretical mathematics from the University of Novi Sad, Serbia in 2015. In 2017, she obtained the M.Sc. degree in mathematics from the University of Bonn, Germany. In September 2017, she started the Ph.D. studies in computer science at the Computer Vision group of the Friedrich Schiller University Jena and the Climate Informatics group of the German Aerospace Institute for Data Science in Jena, where she is working on deep probabilistic graphical models and causal inference with applications to detecting anomalies in biogeochemical processes.

MAHA SHADAYDEH received the Ph.D. degree in Electrical Communications Engineering from Tohoku University, Sendai, Japan in 1999. She was an associate professor at the Electronic Engineering Department, Damascus University, Syria (2000-2011) and a researcher at the Machine Perception Research Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences (2012-2016). In 2007, she was a visiting associate professor at the Multimedia and Signal Processing Laboratory, University of British Colombia, Vancouver, Canada. Since 2017, she is working as a researcher and team leader at the Computer Vision Group, Department of Mathematics and Computer Science, Friedrich Schiller University Jena, Germany. Her research interests comprise remote sensing image analysis, biomedical signal processing, dynamical system analysis, and causal reasoning.



JAKOB RUNGE studied physics at Humboldt University Berlin, Germany and obtained the Ph.D. degree at the Potsdam Institute for Climate Impact Research in 2014. From 2016 to 2017 he was a fellow at Imperial College London, United Kingdom. His awards include the Carl-Ramsauer doctoral prize and a Fellowship Award in Studying Complex Systems by the James S. McDonnell Foundation. In 2020 he won an ERC Starting Grant with his interdisciplinary project CausalEarth. Since 2017, he heads the Climate Informatics group at German Aerospace Center's Institute of Data Science in Jena, Germany since 2017 and is a guest professor of computer science at TU Berlin since 2021.

MARKUS REICHSTEIN received the Ph.D. degree from the department of Plant Ecology at the University of Bayreuth in 2001. He is a Director of the Biogeochemical Integration Department at the Max-Planck-Institute for Biogeochemistry in Jena, Germany. His main research interests revolve around the response and feedback of ecosystems (vegetation and soils) to climatic variability with an Earth system perspective, considering coupled carbon, water and nutrient cycles. Of specific interest is the interplay of climate extremes with ecosystem and societal resilience.

Since 2013, he is Professor for Global Geoecology at the Friedrich Schiller Univerisity Jena, and founding Director at the Michael-Stifel-Center Jena for Data-driven and Simulation Science. He has been serving as lead author of the IPCC special report on Climate Extremes (SREX), as member of the German Commitee Future Earth on Sustainability Research, and the Thuringian Panel on Climate.

JOACHIM DENZLER (member of IEEE computer society (M)) earned the degrees ”Diplom-Informatiker”, ”Dr.-Ing.” and ”Habilitation” from the University of Erlangen, Germany, in years 1992, 1997, and 2003, respectively. Currently, he holds a position as full professor for computer science and is head of the Computer Vision Group at the Friedrich Schiller University Jena, Germany. He is also Director of the Data Science Institute of the German Aerospace Center. His research interests comprise the automatic analysis, fusion, and understanding of sensor data, especially development of methods for visual recognition tasks and dynamic scene analysis.

He contributed in the area of active vision, 3D reconstruction, and object recognition and tracking. He is author and co-author of over 300 journal and conference papers and technical articles. He is a member of DAGM, and GI.