# Reduced order digital twin and latent data assimilation for global wildfire prediction

Caili Zhong[1], Sibo Cheng[2], Matthew Kasoar[3], Rossella Arcucci[1]

[1]Department of Earth Science and Engineering, Imperial College London, London, United Kingdom
[2]Data Science Institute, Imperial College London, London, United Kingdom
[3]Department of Physics, Imperial College London, London, United Kingdom

*Correspondence to*: Sibo Cheng (sibo.cheng@imperial.ac.uk)

**Abstract.** The occurrence of forest fires can impact vegetation in the ecosystem, property, and human health, but also indirectly affect the climate. JULES-INFERNO is a global land surface model, which simulates vegetation, soils, and fire occurrence driven by environmental factors. However, this model incurs substantial computational costs due to the high data dimensionality and the complexity of differential equations. Deep learning-based digital twins have an advantage in handling large amounts of data. They can reduce the computational cost of subsequent predictive models by extracting data features through Reduced Order Modelling (ROM) and then compressing the data to a low-dimensional latent space. This study proposes a JULES-INFERNO-based digital twin fire model using ROM techniques and deep learning prediction networks to improve the efficiency of global wildfire predictions. The iterative prediction implemented in the proposed model can use current-year data to predict fires in subsequent years. To avoid the accumulation of errors from the iterative prediction, Latent data Assimilation (LA) is applied to the prediction process. LA manages to efficiently adjust the prediction results to ensure the stability and sustainability of the prediction. Numerical results show that the proposed model can effectively encode the original data and achieve accurate surrogate predictions. Furthermore, the application of LA can also effectively adjust the bias of the prediction results. The proposed digital twin also runs 500 times faster for online predictions than the original JULES-INFERNO model without requiring High-Performance Computing (HPC) clusters. The implementation code of this study and the developed models are available at https://github.com/DL-WG/Digital-twin-LA-global-wildfire.

Keywords: Deep learning; Digital twin; Data assimilation; Autoencoder; Recurrent neural networks

# 1 Introduction

30 Every year, unwanted wildland fires result in significant economic, social, and environmental impacts on a global scale (Grillakis et al., 2022). In addition, forest fires are especially sudden, extraordinarily destructive, and notably difficult to timely respond (Grillakis et al., 2022). Globally, an average of more than 200,000 forest fires occurs each year, burning more than 1% of the world's forested area (Pais et al., 2021). Wildfires not only destroy vegetation but can also pose risks to life and property, impact human health due to smoke pollution, and feedback on climate change through the release of stored carbon

35 (Kim, 2015; Marlier et al., 2015; Jauhiainen et al., 2012; Lasslop et al., 2019; Ward et al., 2012). Studying the influence of vegetation and climatic factors on wildfire occurrence is critical for anticipating future wildfires and preparing for their possible impacts on ecosystems and society.

Global fire frequency is related to land use, vegetation type, and meteorological factors. Arid land, hot and dry weather, and

40 combustible vegetation are all wildfire risk factors. Therefore, data from validated natural environment models can predict forest fire risk. Numerous valid and relevant models exist, such as Earth System Models (Claussen et al., 2002) and Dynamic Global Vegetation Models (DGVM) (Prentice and Cowling, 2013). The Joint UK Land Environment Simulator - INteractive Fire and Emissions algorithm for Natural envirOnments (JULES-INFERNO) (Best et al., 2011; Clark et al., 2011) is an example of a DGVM. JULES-INFERNO simulates fire burnt area and emissions over time based on geographic features such

45 as population, land usage, and meteorological conditions (Burton, 2019; Mangeon et al., 2016). However, implementing such a DGVM simulation is time consuming due to the complexity of the physical model and the number of geographical features associated. For example, predicting wildfire occurrence over a 100-year climate change scenario takes approximately 17 hours runtime using 32 threads on the JASMIN High-performance Computing (HPC) service (JASMIN, 2022; Lawrence, 2013) (100-year runtime estimated as average from 4 simulations, range 14.3 – 19.7 hours, using Intel Xeon E5-2640-v4 "Broadwell"

50 or Intel Xeon Gold-5118 "Skylake" processors with ~7 GB RAM available per thread). This is further limited by availability and queuing priority of the compute resource. Therefore, building an efficient digital twin as a surrogate model is necessary for accelerating the prediction process.

The digital twin paradigm combines process-based information and data-driven approaches to best estimate complex systems.

55 An extensive range of scientific problems, including medical science (Quilodran-Casas et al., 2022), nuclear engineering (Gong et al., 2022a), and earth system modelling (Bauer al., 2021), use the digital twin paradigm for modelling. Since the 1990s, researchers have studied the relationship between vegetation and wildfire using machine learning and artificial intelligence (Nadeem et al., 2020). Recent research shows that machine-learning-based Reduced Order Modelling (ROM) can effectively reduce the computational cost by constructing surrogate models (Mohan and Gaitonde, 2018). ROM efficiently

60 compresses the raw data into a low-dimensional space using an encoder and then decompresses the data using a decoder. Thus, predictions and simulations can be performed in a low-dimensional latent space before being decoded into a real physical

space. However, predictive models trained using large amounts of data do not necessarily guarantee long-term prediction accuracy. In fact, iterative applications of Sequence-to-Sequence (Seq2seq) forecasting models can lead to error accumulation, resulting in incorrect long-term predictions (Cheng et al., 2022a; Cheng et al., 2022b). Researchers have applied data

65 assimilation (DA) methods to address this challenge. DA methods correct the predicted data by combining simulated data with observations through specific weighting (Gong et al., 2020). Applications of DA in high-dimensional dynamical systems include weather forecasting (Lorenc et al., 2000; Bonavita et al., 2015; Ma et al., 2014), hydrology (Cheng et al., 2020), and nuclear engineering (Gong et al., 2022b). However, wildfire data from the increasing number of meteorological satellites have high data dimensions (Jain et al., 2020). Applying DA operations on full-size data is computationally expensive, if not

70 prohibitively so. Because there are some major challenges in high dimensional dynamical system models, such as ROM, dynamical system identification, and model error correction (Cheng et al., 2023).

Latent data Assimilation (LA), which combines ROM, ML surrogated models, and DA was recently proposed (Peyron et al., 2021) and applied to a wide range of engineering problems, including air pollution modelling (Amendola et al., 2021),

75 multiphase fluid dynamics (Cheng et al., 2022a) and regional wildfire predictions (Cheng et al., 2022b). In LA, data compression happens before the DA operation (Peyron et al., 2021), significantly reducing the computational cost. Iteration of this prediction assimilation process improves the starting point of the next time-level forecast and leads to more robust long-term predictions.

80 The main objective of this study is to propose a digital-twin model with the same burnt area output as the JULES-INFERNO system. The digital twin model combines ROM, machine learning predictive models, and LA. Data from the JULES-INFERNO simulations are used to train the machine-learning-based digital-twin model. Time series of wildfire-related data is given as input to predict the occurrence of wildfires in following years. The proposed model is tested on unseen initial conditions to predict subsequent fire conditions. This digital twin can significantly improve prediction efficiency compared to

85 the physical model.

Two ROM approaches, Convolutional Autoencoder (CAE) and Principal Component Analysis (PCA), are chosen to reduce data dimensionality. Long Short-Term Memory (LSTM) forms the main structure for wildfire occurrence probability prediction because LSTM is suitable for dealing with dynamic simulations with long-term temporal correlations (Graves and

90 Schmidhuber, 2005a). LA is applied to correct the model results as soon as 'observation' data become available. The 'observation' data used here for LA is the encoded data from the original JULES-INFERNO simulations, which we use as a proof-of-concept substitute for high-quality observation data. Compared to the traditional DA in the entire physical space, LA can considerably improve computational efficiency thanks to the ROM (Graves and Schmidhuber, 2005a). This research aims to use the initial months' output from JULES-INFERNO as input to implement a global-scale wildfire predictive model that

95 combines ROM, recurrent neural networks (LSTM), and LA for efficient wildfire forecasting.

To summarize, the main contributions of this study are as follows.

- This research implements a deep learning digital twin for wildfire prediction based on the JULES-INFERNO land-surface model. The digital twin can greatly improve the computational efficiency of the physical model, by embedding the input into a low-dimensional space before prediction.

- We tested and compared two ROM approaches: Convolutional Autoencoder (CAE) and Principal Component Analysis (PCA), in terms of reconstruction and prediction accuracy over the simulation period of 1961 to 1990.The objective is to achieve a significant reduction in data dimensionality to improve the efficiency of subsequent predictions but maintain the original characteristics of the data. The digital twin achieves long-term predictions by making iterative predictions by using the current predicted results as the next-level input to predict the situation in the next time step.

- LA is applied to avoid error accumulation. LA compares the predicted values against the original JULES-INFERNO outputs (considered as observations in this study) and periodically adjusts the predictions. When applying unseen scenarios to the developed model, LA adjusts the predicted values to improve the accuracy of subsequent predictions. Therefore, LA implemented in this project can effectively stabilize prediction results.

The paper is organized into the following sections: Sect. 2 explains the original physical model, JULES-INFERNO, and the data set of this project. Sect. 3 describes the dimensionality reduction method used in the study (PCA and the constructed CAE structure), explains the surrogate model used in this research for wildfire prediction, and outlines the principles and applications of LA. Sect. 4 discusses the experimental results and analysis of the study. Finally, Sect. 5 concludes the most important findings of this research and suggests future directions for enhancement.

## 2    JULES-INFERNO

The Joint UK Land Environment Simulator (JULES) is simulating land surface vegetation, carbon stores, and hydrology, primarily using physical models to simulate the processes of land-use, water and carbon fluxes with the atmosphere, and climate interactions with vegetation dynamics (Best et al., 2011; Clark et al., 2011). The INFERNO fire scheme was constructed based on a simplified parameterization of fire ignition and vegetation flammability (Pechony and Shindell, 2009). Coupled with JULES, the INFERNO scheme predicts fire burnt area and carbon emissions based on the simulated vegetation, as well as vegetation mortality due to fire, which feeds back on the vegetation distribution (Burton, 2019; Mangeon et al., 2016). More precisely, the model calculates flammability based on soil moisture, fuel density, temperature, humidity, and

precipitation, and then combines lightning strikes to estimate the average area burnt. Finally, emitted atmospheric aerosols and trace gases can be calculated from vegetation-dependent emission factors (Mangeon et al., 2016). Figure 1 illustrates the input

130   variables and critical components of JULES-INFERNO. However, this model involves a large amount of data and parameters, leading to high computational cost. In the present paper, we propose the use of a surrogate model that substitutes the original high-precision simulation model. The surrogate model aims to use the same input fields as the original model and outputs a result that approximates the original INFERNO burnt area output but is less computationally intensive to solve.

135   This research considered four meteorological boundary conditions and predicted burnt area (temperature ($T$), humidity ($H$), rainfall ($R$), and lightning ($L$)) and the field of burnt area fraction ($P$)) from the JULES-INFERNO model as input data.  We used an ensemble of five JULES-INFERNO simulations which were forced using meteorological conditions from the FireMIP last glacial maximum (LGM) scenario (Rabin et al., 2017), which uses a detrended 1961-1990 re-analysis timeseries, uniformly shifted to match LGM global average climate, as the atmospheric boundary conditions. Monthly-mean values were collected

140   for each meteorological boundary condition, nominally from 1961-1990 for each simulation. Thus 360 snapshots for each variable are available, denoted as temperature ($T_i, i = 1, 2, \cdots, 360$), humidity ($H_i, i = 1, 2, \cdots, 360$), rainfall ($R_i, i = 1, 2, \cdots, 360$), and lightning ($L_i, i = 1, 2, \cdots, 360$). JULES-INFERNO was run with a resolution of 1.25° latitude $\times$ 1.875° longitude, giving each snapshot a latitude span of 144 units and a longitude span of 192 units, so the size of each snapshot is $144 \times 192$. We denote the wildfire burned area predicted by JULES-INFERNO from each of the 5 ensemble members as

145   $P_s, s = 1, 2, \cdots, 5$). Although each ensemble member simulated the same nominal period (1961 to 1990) and was forced by the same detrended meteorological boundary conditions, we applied different initial internal states to sample a range of model internal variability. As shown in Table 1, $P_1$ was randomized, and the initial internal states of the subsequent experimental results were all the last internal states of the previous one. The burnt area model output is only diagnosed over latitudes with non-zero land cover, and so ($P_{s,i}, s = 1, 2, \cdots, 5, i = 1, 2, \cdots, 360$), only spans 112 latitude units, meaning the fire snapshot size

150   is $112 \times 192$ rather than $144 \times 192$.

In subsequent experiments,  $P_1, P_2$ and portion of $P_3$  are the training set and another portion of $P_3$ is the validation set during training. $P_4$ and $P_5$ were used as the test set for all models built in the study. In addition, to eliminate the adverse effects of odd sample data in training, this research standardized all training and test sets (climates and fire variables) by applying Eq. (1) to

155   normalize the data to [0, 1]:

$$P_s = \frac{P_s - P_{s_{min}}}{P_{s_{max}} - P_{s_{min}}}, s = 1, 2, \cdots, 5, \tag{1}$$

**Table 1 JULES-INFERNO Experiments' initial internal states**

| Experiment | P1 | P2 | P3 | P4 | P5 |
|---|---|---|---|---|---|
| Initial conditions | 1200-year spin-up repeating the 1961-1990 timeseries, originally initialized from an arbitrary present-day JULES run | Jan 1st 1991 from P1 | Jan 1st 1991 from P2 | Jan 1st 1991 from P3 | Jan 1st 1991 from P4 |
| Ignitions | Lightning | Lightning | Lightning | Lightning | Lightning |

160

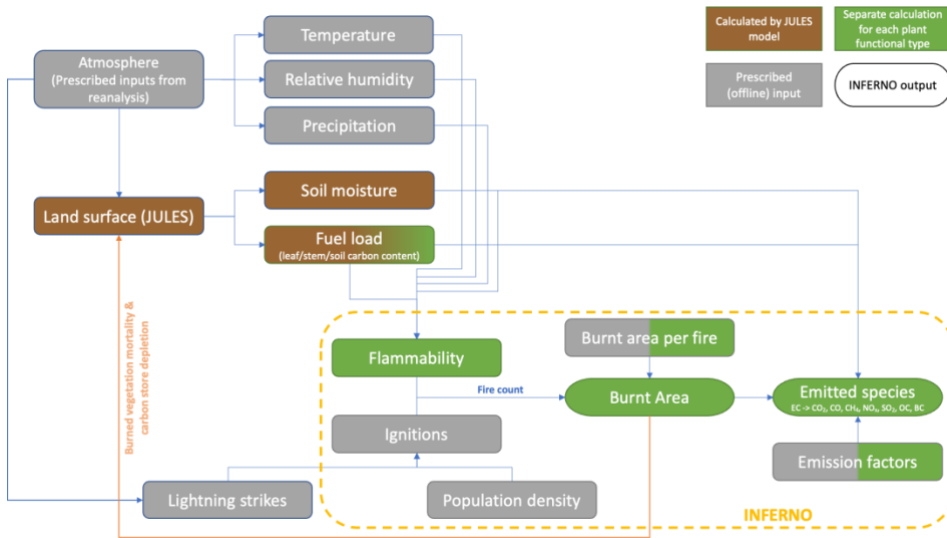



**Figure 1 The mechanism of JULES-INFERNO**

## 3   Methodology

In this section, we present the main technologies used to develop the digital twin: ROM, surrogate predictive model, and LA. The computation of the digital twin starts by constructing the ROM for reducing the dimension of JULES-INFERNO data. The climate and wildfire data are then concatenated and processed into time-series data and fed into the training predictive model. Afterwards, iterative forecasting tests the trained surrogate model. The test uses twelve months of unseen data as input to predict fires for the next 29 years. LA periodically adjusts the forecasting process to ensure stability, accuracy, and sustainability.

## 3.1 Reduced Order Modelling

Assume that the original data is an n-dimensional vector denoted as $U = \{u_i\}_{i=1,2,\cdots,n}$; we then denote the compressed variable (k-dimension, $k < n$) after dimensionality reduction as $\widetilde{U} = \{\tilde{u}_i\}_{i=1,2,\cdots,k}$. The variable that reconstructs the compressed data to its original size is denoted as $\widehat{U} = \{\hat{u}_i\}_{i=1,2,\cdots,n}$. ROM's primary purpose is to minimize the expectation of the Mean Square Error (MSE) of reconstruction, i.e., $E\left[(U - \widehat{U})^2\right]$. The meteorological boundary conditions and burnt area data obtained by dimensionality reduction can therefore be denoted by $\widetilde{T}_\iota, \widetilde{H}_\iota, \widetilde{R}_\iota, \widetilde{L}_\iota, \widetilde{P_{s,\iota}}, s = 1,2,\cdots 5; i = 1,2,\cdots 360$, and each reduced snapshot is k-dimensional. Decoded data are denoted as $\widehat{T}_\iota, \widehat{H}_\iota, \widehat{R}_\iota, \widehat{L}_\iota, \widehat{P_{s,\iota}}, s = 1,2,\cdots 5; i = 1,2,\cdots 360$, and dimensionality of each element is consistent with the original data (n-dimension).

### 3.1.1 PCA

PCA extracts essential information from the data and eliminates redundant information by analysing principal components. PCA uses the idea of dimensionality reduction to form new variables by linearly combining multiple parameter indicators, mapping the original n-dimensional features to k-dimensional features $(n > k)$, k is known as the truncation parameter in PCA (Bianchi et al., 2015).

The compressed variable $\tilde{u}_i$ can be represented by a linear combination of $u_1, u_2, \ldots, u_n$ as shown in Eq. (2), to make the global variance after the transformation maximal.

$$\begin{cases} \tilde{u}_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \cdots + \alpha_{1n}u_n \\ \tilde{u}_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \cdots + \alpha_{2n}u_n \\ \tilde{u}_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \cdots + \alpha_{3n}u_n, \\ \qquad\qquad \vdots \\ \tilde{u}_k = \alpha_{k1}u_1 + \alpha_{k2}u_2 + \cdots + \alpha_{kn}u_n \end{cases} \tag{2}$$

where, $\alpha_i = [\alpha_{i1}, \alpha_{i1}, \cdots, \alpha_{in}]^T$ $(i = 1, 2, \cdots, k)$ is the eigenvector associated to the eigenvalues $\lambda_i$ of the covariance matrix C.

The covariance matrix C is shown in Eq. (3), where the formula for the covariance is shown in Eq. (4).

$$C = \begin{bmatrix} cov(u_1, u_1) & cov(u_1, u_2) & \cdots & cov(u_1, u_n) \\ \vdots & & \ddots & \vdots \\ cov(u_n, u_1) & cov(u_n, u_2) & \cdots & cov(u_n, u_n) \end{bmatrix}, \tag{3}$$

$$cov(u_i, u_j) = E(u_i u_j) - E(u_i)E(u_j) \quad i,j = 1, 2, \cdots, n, \tag{4}$$

where, $E(u_i)$ represents the mathematical expectation of $u_i$.

The contribution rate of each principal component can be calculated using the principal component explained variance formula (Eq. (5)), the principal components corresponding to the special eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ are selected in the descending order to construct $\widetilde{U}$. The cumulative explained variance is referred to Eq. (6).

$$\alpha_i = \frac{\lambda_i}{\sum_{j=1}^{k} \lambda_j}, i, j = 1, 2, \cdots, k, \tag{5}$$

$$\sum_{i=1}^{k} \alpha_i = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{k} \lambda_j}, i, j = 1, 2, \cdots, k, \tag{6}$$

### 3.1.2 CAE

200   An autoencoder (AE) is a self-supervised deep learning method that minimizes the reconstruction error between its model input and output. The autoencoder optimizes parameters to obtain a low-dimensional data representation of high-dimensional features (Huang et al., 2017). The feature extraction method used in standard AE is fully connected and straightforward to implement. However, each neuron connects to all the neurons in the next layer. Thus, standard AE generates massive parameters, making the computation more expensive while ignoring some spatial patterns in the image (Masci et al., 2011).

205   Convolutional Autoencoder (CAE) (Masci et al., 2011), a combination of AE and CNN, has been proposed to address the drawbacks of fully connected AEs. The CAE model inherits the self-supervision function of standard AE but replaces the matrix product operation between hidden neurons with convolution and pooling operations. Compared to AE, CAE can capture local spatial patterns of monitoring data (Masci et al., 2011). Furthermore, extracting features by convolution can reduce the number of parameters and increase the training speed.

210

This study includes climate and fire conditions from various regions in the form of two-dimensional images. Therefore, this research constructed a CAE model using AE reconstruction, dimensionality reduction features, and CNN local feature extraction capabilities. The CAE model aims to achieve self-supervised learning of environmental and wildfire features.

215   The encoder designed for this study uses three convolutional layers, three max-pooling layers and two fully connected layers. The decoder includes four convolutional layers, three up-sampling layers and one fully connected layer. Figure 2 shows the CAE structure of this study with a latent space of dimension 100, and Conv2d, MaxPooling2D, Dense, UpSampling2D representing the 2D convolutional layer, max pooling layer, fully connected layer, and up-sampling layer respectively.

220   Firstly, the sample data, $U = \{u_i\}_{i=1,2,\cdots,n}$ need to be converted into a two-dimensional matrix of $p$ rows and $q$ columns ($p * q = n$) according to the latitude and longitude range of the original data, and then each snapshot could be interpreted as an image. Given the input $U$, features are extracted by multiple convolutional kernels to obtain the output $C_l$ of the $l^{th}$ layer, then the $C_l'$ is obtained by down-sampling through the subsequent pooling layer to retain the main features of the inflow data and prevent over-fitting. The operational methods are shown in Eq. (7) and Eq. (8).

225
$$\begin{cases} C_l = f(W_l \otimes U + b_l), & l = 1 \\ C_l = f(W_l \otimes C_{l-1}' + b_l), & l = 2,3 \end{cases} \tag{7}$$

$$C_{l-1}' = Maxpooling(C_l), \ l = 1,2,3, \tag{8}$$

where: $W_l$ and $b_l$, are the weights and biases of the layer $l$ network respectively; $f(\cdot)$ is the ReLU function; and $Maxpooling(\cdot)$ is the maximum pooling operation.

230 As shown in Figure 2, during the network training process, the encoder acquires the feature mapping of the input data layer by layer. It obtains the k-dimensional latent space data after the Flatten and Dense layers. In contrast, the decoder inputs the latent space data and reconstructs the data by convolution, up sampling, etc., and evaluates the reconstruction performance of CAE on climate and wildfire data by the loss function (MSE). Eq. (9) shows the square error of one sample ($L(\cdot)$) and mean error for the training set ($J(\cdot)$), supposing the training set is $P_{tr}$ with $N_{tr}$ snapshots.

235
$$\begin{cases} L(U,\widehat{U}) = (U - \widehat{U})^2 \\ J(P_{tr},\widehat{P_{tr}}) = \frac{1}{N_{tr}}\sum_{i=1}^{N_{tr}} L(P_{tr,i},\widehat{P_{tr,i}}) \end{cases}, \tag{9}$$

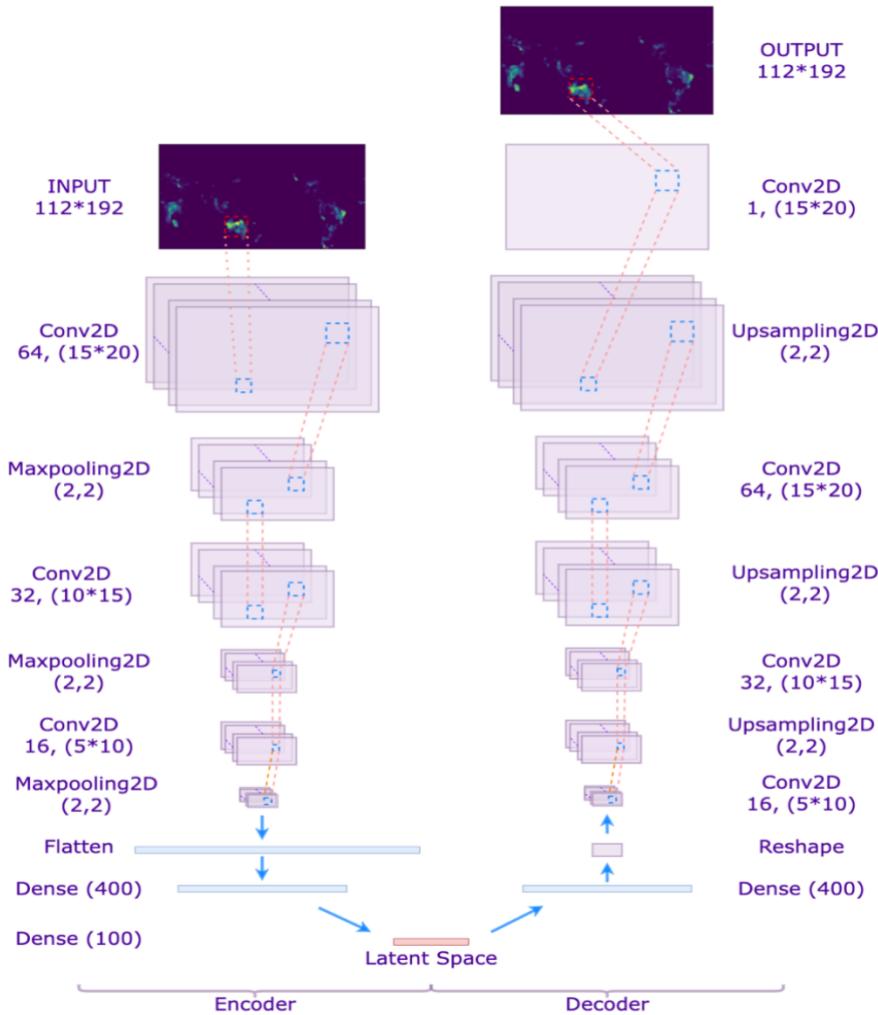



**Figure 2 The CAE Model structure**

## 3.2 Surrogate predictive model

After data compression, the proposed digital twin model combined the climate and fire data as inputs for the predictive
240 surrogate model, denoted as $Z = \{Z_s\}_{s=1,2,\cdots 5}$, and $Z_s = \{Z_{s,i}\}_{s=1,2,\cdots 5;\ i=1,2,\cdots 360} = [\widetilde{P_{s,i}}\ \widetilde{T_i}\ \widetilde{H_i}\ \widetilde{R_i}\ \widetilde{L_i}]$, where 's=1,2,..5' indicates
the 5 fire data sets (I.e., JULES-INFERNO simulations) respectively. The data are normalised before it is fed into the model.

Here we aim to use ML techniques to surrogate the original physical predictive model. The Recurrent Neural Network (RNN)
is a reference method for time series prediction problems. The LSTM network, a variant of RNN, constructed the wildfire
245 predictive model in the reduced latent space using the encoded JULES-INFERNO climate and fire data as inputs. Figure 3
shows the basic structure of the whole surrogate model.

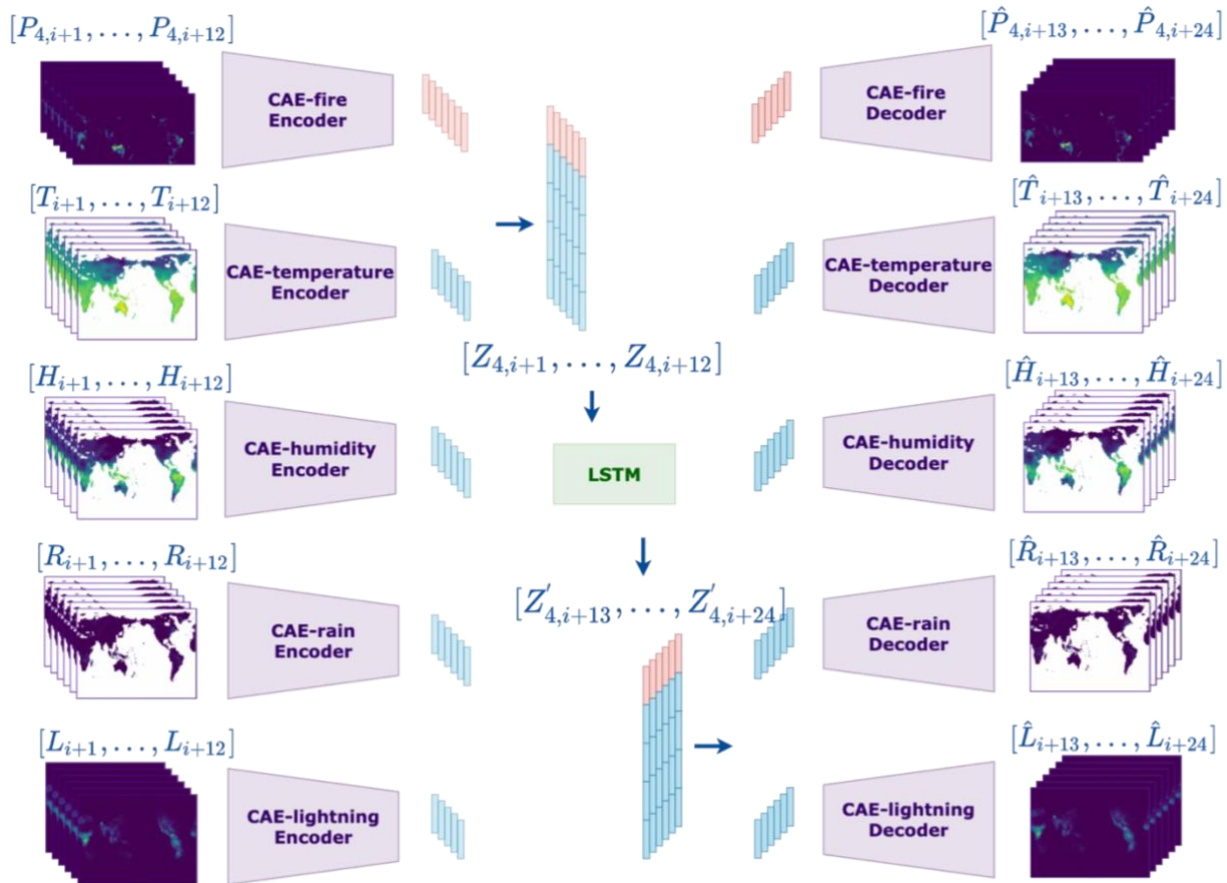



**Figure 3 The Surrogate Model structure**

The seq2seq prediction based on the LSTM network is implemented in this study with $t_{in}$ timesteps as input and $t_{out}$ timesteps
250 as output. Here, each timestep represents a month of simulation time. The length of the input and output sequences is fixed as

$t_{in} = t_{out} = 12...$ $Z_{train} = Z_1 \cup Z_2 \cup Z_3$ and the test set consists of $Z_{test} = Z_4 \cup Z_5$. The input of the LSTM training can be obtained by shifting the initial time of each time series, as shown in Eq. (10). The model uses MSE loss function to evaluate the training performance.

$$
\begin{aligned}
\left[Z_{1,1}, Z_{1,2}, \cdots, Z_{1,t_{in}}\right] &\xrightarrow{LSTM\ train} \left[Z_{1,t_{in}+1}, Z_{1,t_{in}+2}, \cdots, Z_{1,t_{in}+t_{out}}\right], \\
\left[Z_{1,2}, Z_{1,3}, \cdots, Z_{1,t_{in}+1}\right] &\xrightarrow{LSTM\ train} \left[Z_{1,t_{in}+2}, Z_{1,t_{in}+3}, \cdots, Z_{1,t_{in}+t_{out}+1}\right] \\
&\vdots \\
\left[Z_{1,360-t_{in}-t_{out}+1}, \cdots, Z_{1,360-t_{out}}\right] &\xrightarrow{LSTM\ train} \left[Z_{1,360-t_{out}+1}, Z_{1,360-t_{out}+2}, \cdots, Z_{1,360}\right] \\
&\vdots \\
\left[Z_{3,360-t_{in}-t_{out}+1}, \cdots, Z_{3,360-t_{out}}\right] &\xrightarrow{LSTM\ train} \left[Z_{3,360-t_{out}+1}, Z_{3,360-t_{out}+2} \cdots, Z_{3,360}\right].
\end{aligned}
\tag{10}
$$

255

As for the iterative predictions in test dataset, the trained model uses the current predictive result as the input of next prediction. The output of the predictive model can be denoted as the $Z'_{test} = Z'_4 \cup Z'_5$. The first-year data of the test set is used as model input, and the prediction result set only contains data from the following 29 years. At each iteration, the model can predict the global wildfire risk for the next year, as shown in Eq. (11).

$$
\begin{aligned}
\left[Z_{4,1}, Z_{4,2}, \cdots, Z_{4,12}\right] &\xrightarrow{LSTM\ predict} \left[Z'_{4,13}, Z'_{4,14}, \cdots, Z'_{4,24}\right], \\
\left[Z'_{4,13}, Z'_{4,14}, \cdots, Z'_{4,24}\right] &\xrightarrow{LSTM\ predict} \left[Z'_{4,25}, Z'_{4,26}, \cdots, Z'_{4,36}\right] \\
&\vdots \\
\left[Z'_{4,337}, Z'_{4,338}, \cdots, Z'_{4,348}\right] &\xrightarrow{LSTM\ predict} \left[Z'_{4,349}, Z'_{4,350}, \cdots, Z'_{4,360}\right], \\
\left[Z_{5,1}, Z_{5,2}, \cdots, Z_{5,12}\right] &\xrightarrow{LSTM\ predict} \left[Z'_{5,13}, Z'_{5,14}, \cdots, Z'_{5,24}\right] \\
&\vdots \\
\left[Z'_{5,337}, Z'_{5,338}, \cdots, Z'_{5,348}\right] &\xrightarrow{LSTM\ predict} \left[Z'_{5,349}, Z'_{5,350}, \cdots, Z'_{5,360}\right].
\end{aligned}
\tag{11}
$$

260

Thanks to the gate structure, LSTM networks are efficient in dealing with long-term temporal correlations that standard RNN cannot handle (Graves and Schmidhuber, 2005b). More precisely, the essence of the gate structure of the LSTM is to use the sigmoid activation function so that the fully connected network layer outputs a value between 0 and 1, describing the proportion of the information quantity passed. The forgetting gate indicates the proportion of the output information quantity forgotten at the last moment, and the input gate represents the proportion of the input information quantity retained at the current moment, both updating the state value together. Finally, the output gate represents the proportion of the new state output. Suppose $x_t$ ($x_0 = \left[Z_{s,i}, Z_{s,i+1}, \cdots, Z_{s,i+t_{in}-1}\right], s = 1,2,3, i = 1,2, \cdots, 349$) are the current inputs, $h_{t-1}$ is the output of last moment, $c_t$ is the value of new state, $c_{t-1}$ is the state value of last moment. Equations (12, 13, 14, 15, 16,17) represent the inner principal of the LSTM.

265

270

Input state: $z_t = \tanh(W_z x_t + Q_z h_{t-1} + b_z),$ (12)

Input gate: $i_t = \sigma(W_i x_t + Q_i h_{t-1} + b_i),$ (13)

Forget gate: $f_t = \sigma(W_f x_t + Q_f h_{t-1} + b_f),$ (14)

Current state: $c_t = f_t * c_{t-1} + i_t * z_t,$ (15)

Output gate: $o_t = \sigma(W_o x_t + Q_o h_{t-1} + b_o),$ (16)

Current output: $h_t = o_t * \tanh(c_t),$ (17)

where, $W_z, W_i, W_f, W_o$ are input weighting matrix; $Q_z, Q_i, Q_f, Q_o$ are loop weights; $b_z, b_i, b_f, b_o$ are bias value, and $\sigma(\cdot)$ is the sigmoid function.

## 3.3 Latent Data Assimilation

The basic principle of DA is the combination of numerical models with observation data to improve the forecast of the system under study (Vallino, 2000). When applying the developed predictive model to unseen initial conditions, we apply the DA method periodically to enhance the current prediction results with the help of the observation data. The observation data in this research is considered the original output of the JULES-INFERNO model so that we can simulate the situation where high-quality real-time observation data are available. As periodic corrections are made to the forecast data, DA allows for a more stable and accurate long-term prediction. In this study, LA was applied, which combines ROM with DA, i.e., reducing the dimensionality of the data before applying the DA method (Peyron et al., 2021). In addition, ROM could reduce the parameters for subsequent operations, which allows LA to have a significant advantage in terms of computational efficiency compared to the classical full-space DA.

Both DA and LA can be summarized as a problem of solving the minimization of an objective function $J_{DA}(\cdot)$ characterizing the deviation between the analysis field (optimal state $\dot{Z}_{test}$) and the observation field (actual state $Z_{test}$) as well as the background field (predicted state $Z'_{test}$), as shown in Eq. (18).

$$J_{DA}(\dot{z}_t) = \frac{1}{2}(\dot{z}_t - z'_t)^T B^{-1}(\dot{z}_t - z'_t) + \frac{1}{2}[z_t - \mathcal{H}(\dot{z}_t)]^T O^{-1}[z_t - \mathcal{H}(\dot{z}_t)],$$ (18)

where $t$ indicates the temporal index (the $t^{th}$ predicted year), $\dot{z}_t$ is the analysis variable of DA, $z'_t$ is the background variable, $z_t$ is the observation (original) variable. B and O represent the background and observation error covariance matrix. The modelling of these covariances determines the weight of prediction and observation in the objective function, which is crucial for DA approaches (Lawless et al., 2005). $\mathcal{H}(\cdot)$ is the state-observation function in latent space, which maps the analysis variable to the observation variable. In this study, since we are assimilating the compressed output of the original JULES-INFERNO model, $\mathcal{H}(\cdot)$ is the identity function.

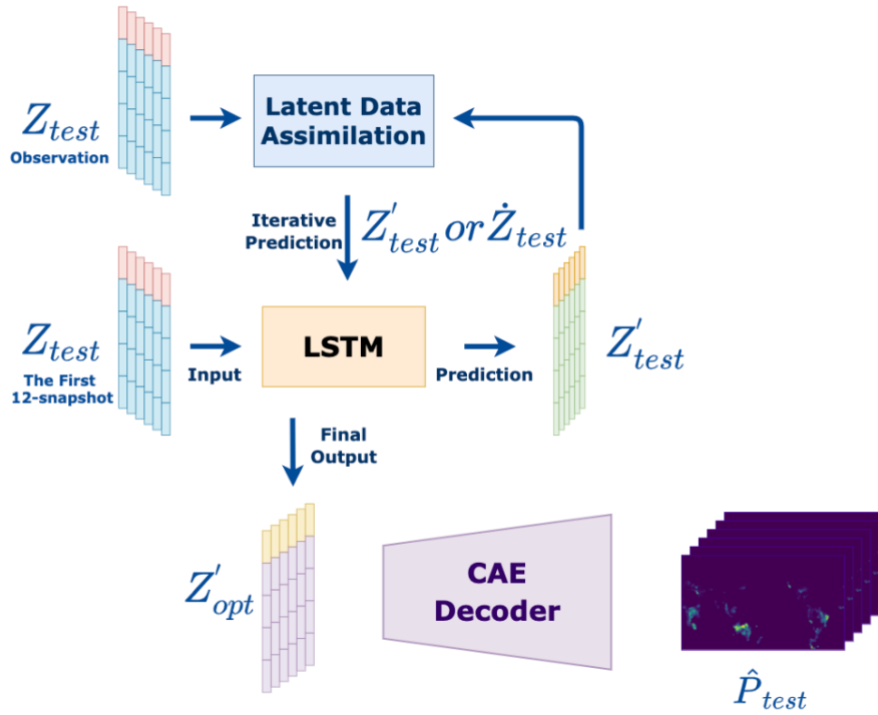
The latent analysis field $\dot{Z}_{test,t}$ could be obtained by optimizing the objective function (Eq. (18)),

$$\dot{Z}_{test,t} = argmin(J_{DA}(\dot{z}_t)),$$ (19)

305 The Eq. (18) can be solved by the Best Linear Unbiased Estimator (Fulton, 2000), since the transformation operator $\mathcal{H} = H$ is a linear function in the latent space,

$$\dot{Z}_{test,t} = z'_t + BH^T(HBH^T + O)^{-1}(z_t - \mathcal{H}(z'_t)). \tag{20}$$

The obtained analysis state $\dot{Z}_{test,t}$ can be used as an initial point for the next level prediction, as shown in Figure 4, the predicted

310 field in the latent space could be $\dot{Z}_{test,t}$ (when applying LA at $t$ time index) or $Z'_{test,t}$ ( LSTM prediction at $t$ time index). These processes can be repeated periodically to consistently improve the prediction performance. Furthermore, in Figure 4, the final predicted sequence of the digital twin is the combination of the $\{\dot{Z}_{test,t}\}_{n\epsilon[0,29]}$ and $\{Z'_{test,m}\}_{m\in[0,29]}$ $(m \neq n)$, which could be denoted as $Z'_{opt}$.

315 **Figure 4 The Predict Model with LA**

## 4 Numerical Results

In this section, we evaluate the proposed digital twin model's performance regarding prediction accuracy and computational efficiency when the digital twin model faces unseen initial conditions.

**4.1 ROM results**

320 To set an optimal dimension of the latent space, we examined the explained variance of PCA for the training dataset. The training dataset consists of P1, P2, and P3 with different numbers of principal components, as shown in Figure 5. According to the work of (Li et al., 2021), 85% to 95% of the explained variance can reflect most data information while effectively reducing the original data dimensionality. Therefore, to reduce the latent space dimension and the computational cost of subsequent processing, we chose in this research the 100-dimension latent space with 87% explained variance, as shown in

325 Figure 5. In addition, we also chose the 20-dimensional latent space for both PCA and CAE for comparison purposes.

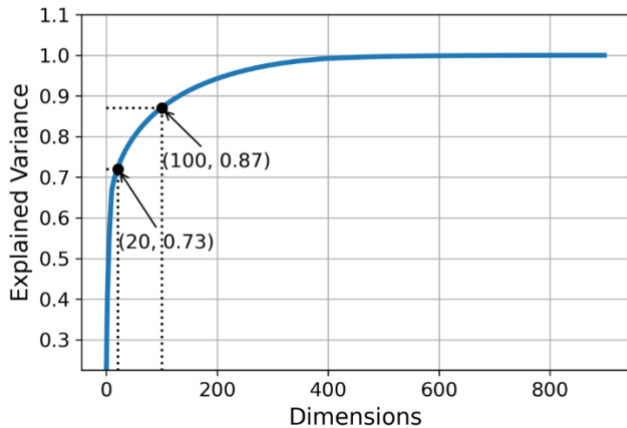



**Figure 5 PCA Explained Variances with different Dimensions**

In terms of performance evaluation, the scenarios $P_1 \cup P_2 \cup P_3$ are used as the training set and $P_4 \cup P_5$ are considered as the test set. The number of training epochs for the CAE models is fixed as 1000. The MSE between reconstructed and original

330 simulations for different models are shown in Table 2. To quantify the mismatch between the JULE-INFERNO output $P$ and decoded prediction $P'$, we compute the MSE averaged by the number of pixels, i.e.,

$$\epsilon^{MSE} = \sqrt{\sum ||P_{i,j} - P'_{i,j}||_2^2 / (112 * 192)}$$

**Table 2 PCA and CAE performances**

| CAE | Latent Space | Test Set | average MSE Loss |
|---|---|---|---|
| PCA | 20 | P4+P5 | 0.000245768 |
| PCA | 100 | P4+P5 | 0.000176458 |
| CAE | 20 | P4+P5 | 0.000213044 |
| CAE | 100 | P4+P5 | 0.000164788 |

According to Table 2, for both PCA and CAE with 100-dimensional latent space, the reconstruction error is significantly lower compared to 20-dimensional. Furthermore, it can be evaluated from the experimental results that the CAE reconstruction MSE is lower compared to PCA. As previously described, CAE can capture feature information in 2-dimensional images during the encoding process, while PCA, which performs compression in 1-dimension space, may ignore these spatial patterns. In summary, CAE with 100-dimensional latent space has the best reconstruction performance in these experiments. As for the compression of climatic conditions $T, H, R \ and \ L$, 83% data are used for PCA and CAE model training, and the rest of the data are used for model testing, where PCA and CAE have a similar performance.

## 4.2 Predictive model and Latent Data Assimilation

After completing the ROM, the original JULES-INFERNO data are compressed into specific latent spaces, considerably reducing the number of parameters in the predictive model. The concatenated wildfire burnt area field and the four climate variables act as inputs to the surrogate model. Thus, the surrogate model achieves the same inputs as JULES-INFERNO. The predictive model (LSTM) trains separately using encoded data from different ROMs. Similar to the ROM, $P_1$, $P_2$ and $P_3$ are used as the training set while $P_4$, $P_5$ are used as the test set to evaluate the model performance. Each LSTM model has been trained 10000 epochs. As explained in section 3.2, iterative predictions are used for the proposed digital twin. In other words, we first input the first year (12 snapshots) data and then use the current predicted result for the next-level prediction, until the $30^{th}$ year's total burned area is predicted. The MSE for different combinations of ROM and LSTM are shown in Table 3.

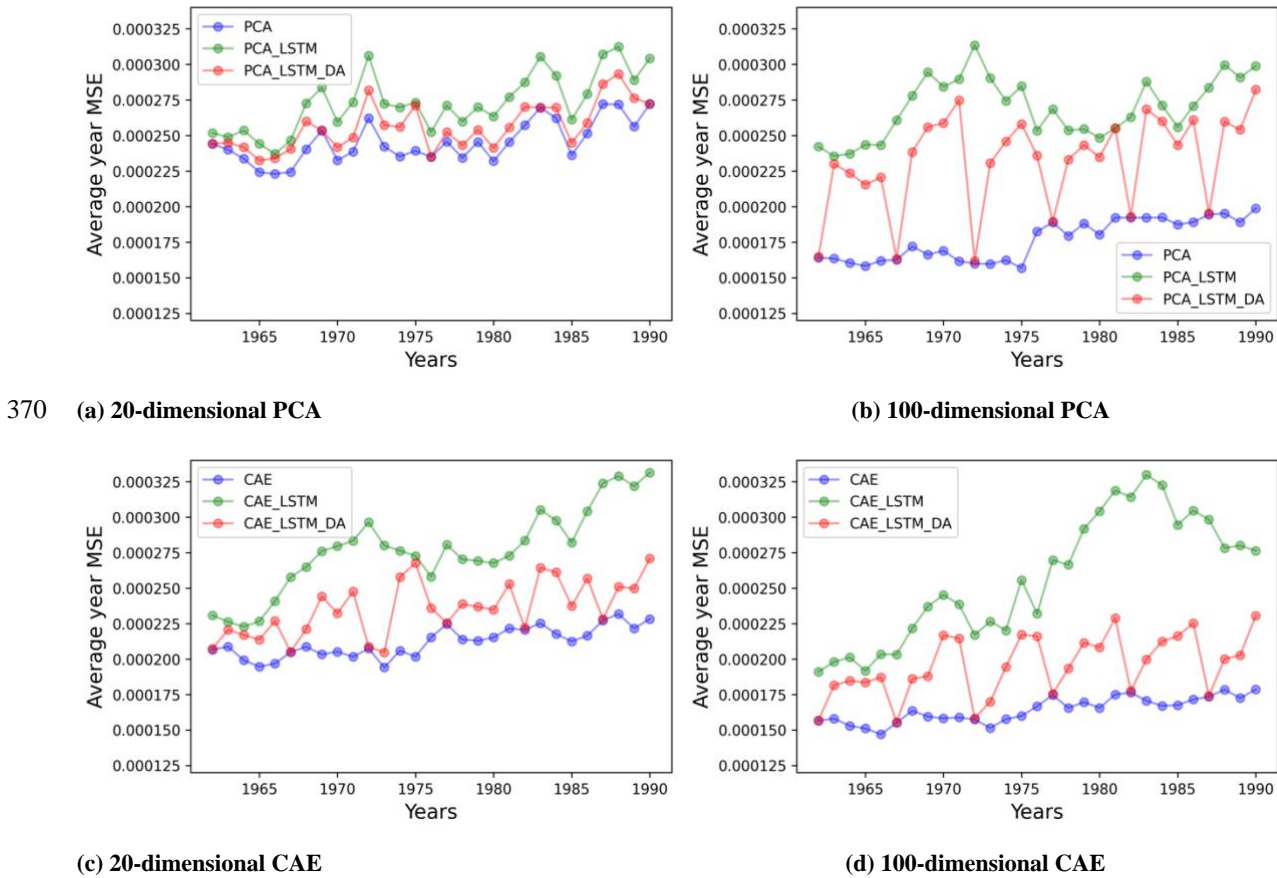
**Table 3 Surrogate Models' performance**

| ROM | Latent Space | Test Set | LSTM Prediction MSE Loss (avg) | LSTM with LA MSE Loss (avg) |
|---|---|---|---|---|
| PCA | 20 | P4+P5 | 0.000272979 | 0.000256832 |
| PCA | 100 | P4+P5 | 0.000273276 | 0.000239031 |
| CAE | 20 | P4+P5 | 0.000276496 | 0.000237333 |
| CAE | 100 | P4+P5 | 0.000256757 | 0.000196481 |

Table 3 shows that the predictions obtained using PCA down to 20 and 100 dimensions and CAE reduced to 20 dimensions are similar, with an MSE close to 2.7E-4. Same as ROM, the best performance in LSTM prediction is obtained by applying CAE with 100-dimensional latent space, with an MSE loss of approximately 2.5E-4.

Comparing the results shown in Table 2 and Table 3, there is a gap between the MSE of the predictive model and the ROM reconstruction, mainly because of the accumulation of errors during the iterative prediction process. Therefore, we periodically used the observations (raw data) and LA to adjust the forecast results to stabilize subsequent forecasts. LA is implemented

every five years during the prediction phase in this study. According to Table 3, there is a steady reduction in forecast MSE after correction by LA. The most significant reduction of MSE is obtained with 100-dimensional CAE compared to other approaches.

To better illustrate the evolution of the prediction error, we display the MSE of the prediction against time in different predictive models. In this study, the existing climate and fire data in 1961 (provided by JULES-INFERNO) with different initial conditions is given to predict the subsequent 29 years of fire. Figure 6 shows the MSE per prediction for the different surrogate models (in red when without LA and in green when with LA) evaluated on $P_4$ test set and we plotted the ROM MSE (in blue) for comparison.

(a) 20-dimensional PCA

(b) 100-dimensional PCA

(c) 20-dimensional CAE

(d) 100-dimensional CAE

**Figure 6 Evolution of MSE against the years in different Surrogate models**

According to Figure 6, using LA can effectively reduce the prediction error, as the vast majority of red points are significantly lower than the corresponding green points. Furthermore, CAE-based surrogate models can better adapt the LA, stabilizing the predictions afterwards and reducing the accumulation of errors. When applying the LA to the PCA-based surrogate models to reduce the error after the simulation, there may still be a sharply increasing forecast error in the subsequent year, as seen in

Figure 6 (a, b). On the other hand, CAE-based models manage to maintain the improvement of DA for future time steps, as shown in Figure 6 (c, d). These results demonstrate the advantage of CAE compared to PCA in terms of generalizability when applied to unseen data. This effect has also been highlighted in the work of (Peyron et al., 2021). Comparing the four cases presented in Figure 6, the 100-dimensional CAE demonstrates the strength of both the original prediction and the assimilated one.

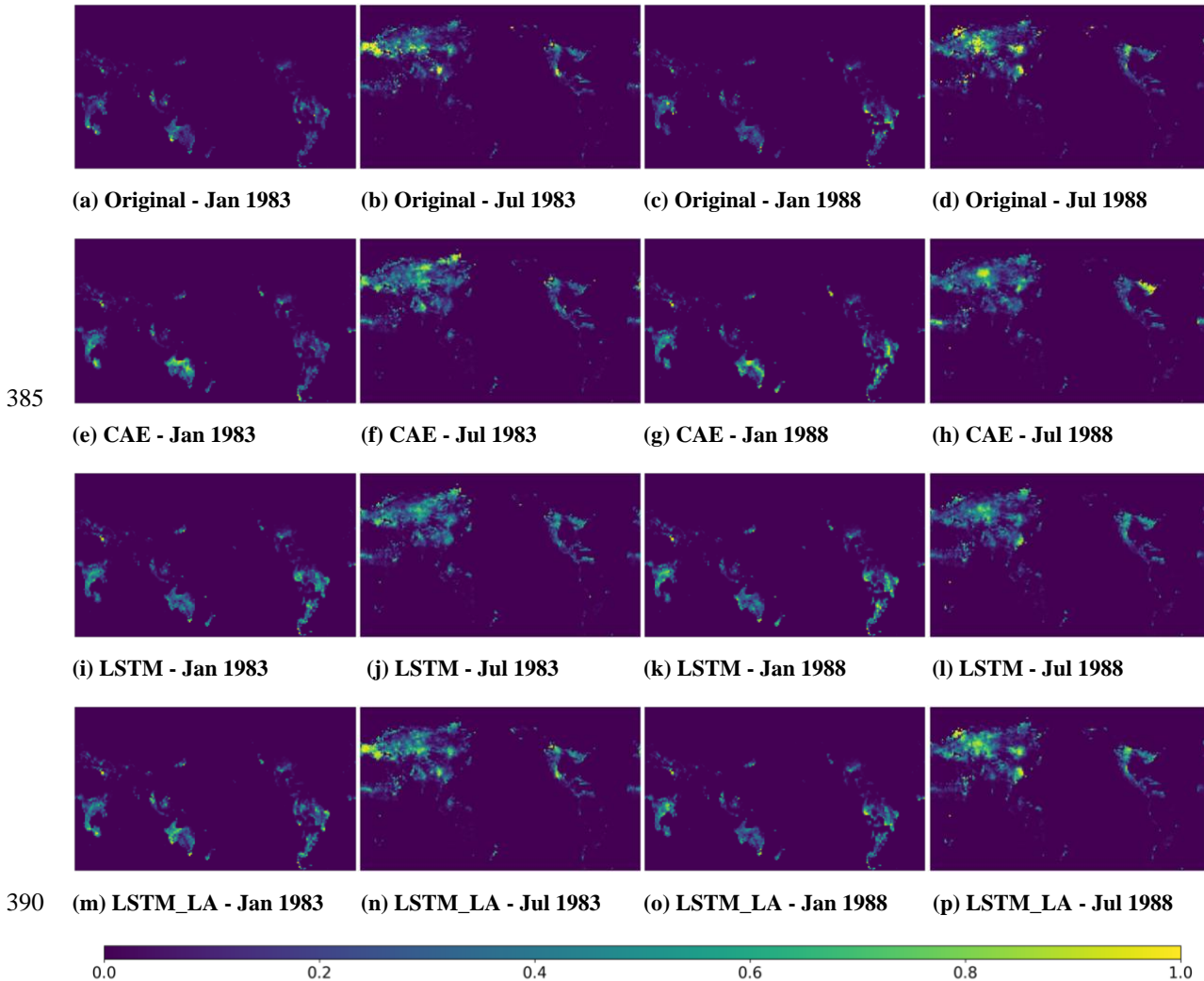



**Figure 7  The nomadized output of 100-dimensional CAE-based digital twin**

Figure 7 displays the output of the100-dimensional CAE-based digital twin model of year 1983 and 1988 (one year after LA is implemented), the assimilated results (m, n, o, p) are compared with the original simulation (a, b, c, d), the CAE reconstruction (e, f, g, h), and the decoded image predicted by the LSTM model (i, j, k, l). In addition, the total burnt area shown in Figure 7 are normalized to the range 0 to 1. The horizontal and vertical coordinates in Figure 7 represent latitude and longitude, and the colour values in the figure represent the grid box which indicates estimated burnt area fraction. The brighter

the colour, the larger the value, colour bar is shown in the bottom of Figure 7. It can be seen from Figure 7 that forest wildfires in January are mainly in the southern hemisphere, such as Oceania in (a) and South America in (c). On the other hand, the forest fires in July are mainly in the northern hemisphere, such as continental Europe in Figure 7 (b, d).

As observed in Figure 7, CAE and LSTM without LA can effectively reconstruct most of the features in the original image. Furthermore, comparing the LSTM predicted images (i, j, k, l) and the optimisation results after applying LA (m, n, o, p), it is found that the assimilation results are significantly closer to the JULE-INFERNO output.

Overall, in this research, we implemented four ROMs based on PCA and CAE, and the comparison reveals that the 100-dimensional CAE has the best reconstruction performance. Then, after combining the ROMs with predictive models (LSTM), a gap between the prediction results and the ROM reconstruction results is noticed. Eventually, the predictions of the surrogate model are made more stable after regular applications of LA to adjust the prediction outputs.

Table 4 compares the prediction time costs of the surrogate models using PCA and CAE downscaled to a 100-dimensional latent space without and with LA, respectively. The prediction time of JULES-INFERNO on HPC is also shown as a reference. As can be seen from Table 4, the difference of PCA and CAE on the running time of the models is not significant; the application of LA slows down the prediction, but still reduces the computing time cost significantly (around 500 times) compared to the original JULES-INFERNO fire predictive model.

**Table 4 Time efficiency comparison**

| Online Computational Time | PCA-100 surrogate | CAE-100 surrogate | PCA-100 LA | CAE-100 LA | JULES - INFERNO |
|---|---|---|---|---|---|
| | $3.41s$ | $2.96s$ | $35.09s$ | $36.62s$ | $\approx 5\ hours$ |

In this study, the training of CAE and LSTM is performed with one NVIDIA P100 GPU (16 Gb VRAM) of Google Colab environment. All the online computation, including decoding, prediction and assimilation, is performed with Google Colab Intel CPUs as shown in Table 4. It is important to highlight that the application of such digital twin doesn't require any HPC or GPU resources. Thus, a personal laptop can also run the model successfully.

## 5    Conclusion and Future work

This study implemented a deep learning-based digital twin for global wildfire prediction using the same input and output fields to replace the physical wildfire model JULES-INFERNO. The proposed model builds ROMs to encode the data into latent space to reduce subsequent processes' computational costs. Then it makes use of the compressed data to train a predictive

425 model based on RNN. Finally, it applies LA to stabilise the prediction results when using the predictive model for iterative prediction to avoid error accumulation. According to the numerical results, the digital twin built using CAE is more accurate compared to the ones using PCA and can effectively capture the features of the original image for encoding and decoding. Applying LA periodically to optimise the prediction results can address the issue of error accumulation and achieve stable long-term predictions. Ultimately the research achieved a much more efficient wildfire-prediction digital twin based on

430 JULES-INFERNO. The proposed model takes only 35 seconds on a laptop to predict 30 years of burnt area fractions, compared to ~5 hours using the original physical model on a 32-thread HPC node. The present research clearly brings additional insight to the computing of reduced order digital twins for high-dimensional dynamical systems. Future work can consider to further improve the generalizability of the proposed approach, by, for instance, training the model with scenarios of different time periods or fine tuning the model when the initial conditions are outside the range of the training set. In addition, it is reported

435 in recent works (Teckentrup et al., 2019) that long-term predictions of JULES-INFERNO can introduce forecast bias. Further efforts can be considered to apply latent data assimilation framework developed in this paper with real-time satellite observations.

**Declaration of competing interests**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared

440 to influence the work reported in this paper

**Acknowledgements**

The authors would like to thank Mr. Zhengdan Shang at Imperial College London for the proofreading of this article. This research is funded by the Leverhulme Centre for Wildfires, Environment and Society through the Leverhulme Trust, grant number RC-2018-023. This work is partially supported by the EP/T000414/1 PREdictive Modelling with QuantIfication of

445 UncERtainty for MultiphasE Systems (PREMIERE). JULES-INFERNO simulations were carried out using JASMIN, the UK's collaborative data analysis environment (https://jasmin.ac.uk).

**References**

Grillakis, M, Voulgarakis, A., Rovithakis, A., Seiradakis, KD., Koutroulis, A., Field, RD., Kasoar, M., Papadopoulos, A. and Lazaridis, M.: Climate Drivers of Global Wildfire Burned Area, 17, https://doi.org/10.1088/1748-9326/ac5fa1, 2022.

450 Pais, C., Miranda, A., Carrasco, J. and Shen, Z. M.: Deep Fire Topology: Understanding the role of landscape spatial patterns in wildfire susceptibility, Environmental Modelling & Software, 143, 105-122, https://doi.org/10.1016/j.envsoft.2021.105122, September 2021.

Kim, S.: Particulate Matter and Ozone: Remote Sensing and Source Attribution, ProQuest Dissertations Publishing, 2015.

Marlier, E. M., DeFries, S. R., Kim, S. P., Koplitz, N. S., Jacob, J. D., Mickley, J. L. and Myers, S. S.: Fire emissions and regional air quality impacts from fires in oil palm, timber, and logging concessions in Indonesia, Environmental Research Letters, 10, 85005, https://dx.doi.org/10.1088/1748-9326/10/8/085005, 2015.

Jauhiainen, J., Hooijer, A. and Page, S. E.: Carbon Dioxide Emissions from an Acacia Plantation on Peatland in Sumatra, Indonesia, Biogeosciences, 9, 617–630, https://doi.org/10.5194/bg-9-617-2012, 2012.

Lasslop, G., Coppola, A.I., Voulgarakis, A., Yue, C. and Veraverbeke, S.: Influence of Fire on the Carbon Cycle and Climate, Current climate change reports, 5, 112–123, https://doi.org/10.1007/s40641-019-00128-9, 2019.

Ward, D. S., Kloster, S., Mahowald, N. M., Rogers, B. M., Randerson, J. T. and Hess, P. G.: The changing radiative forcing of fires: global model estimates for past, present and future, Atmospheric chemistry and physics, 12, 10857–10886, https://doi.org/10.5194/acp-12-10857-2012, 2012.

Claussen, M., Mysak, L., Weaver, A., Crucifix, M., Fichefet, T., Loutre, M. F., Weber, S., et al.: Earth system models of intermediate complexity: closing the gap in the spectrum of climate system models, Climate Dynamics, 18, 579–586, https://doi.org/10.1007/s00382-001-0200-1, 2002.

Prentice, C. I. and Cowling, A. S.: Dynamic Global Vegetation Models, Encyclopedia of Biodiversity, 2, 670-689, https://doi.org/10.1016/B978-0-12-384719-5.00412-3, 2013.

Best, M. J., Pryor, M., Clark, D. B., Rooney, G. G., Essery, R. L. H., Ménard, C. B., Edwards, J. M., Hendry, M. A., Porson, A., Gedney, N., Mercado, L. M., Sitch, S., Blyth, E., Boucher, O., Cox, P. M., Grimmond, C. S. B. and Harding, R. J.: The Joint UK Land Environment Simulator (JULES), model description – Part 1: Energy and water fluxes, Geoscientific Model Development, 4, 677–699, https://doi.org/10.5194/gmd-4-677-2011, 2011

Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., Pryor, M., Rooney, G. G., Essery, R. L. H., Blyth, E., Boucher, O., Harding, R. J., Huntingford, C., and Cox, P. M.: The Joint UK Land Environment Simulator (JULES), model description – Part 2: Carbon fluxes and vegetation dynamics, Geosci. Model Dev., 4, 701–722, https://doi.org/10.5194/gmd-4-701-2011, 2011.

Burton, C., Betts, R., Cardoso, M., Feldpausch, T., Harper, A., Jones, C., Kelley, D., Robertson, E. and Wiltshire, A.: Representation of fire, land-use change and vegetation dynamics in the Joint UK Land Environment Simulator vn4.9 (JULES), Geoscientific Model Development, 12, 179-193, https://doi.org/10.5194/gmd-12-179-2019, 2019.

Mangeon, S., Voulgarakis, A., Gilham, R., Harper, A., Sitch, S. and Folberth, G.: INFERNO: a fire and emissions scheme for the Met Office's Unified Model, Geoscientific Model Development Discussions, 9, 2685–2700, https://doi.org/10.5194/gmd-2016-32, 2016.

JASMIN Site: https://jasmin.ac.uk/, last access: 26 October 2022.

Lawrence, B. N., Bennett, V. L., Churchill, J., Juckes, M., Kershaw, P., Pascoe, S., Pepler, S., Pritchard, M. and Stephens, A.: Storing and manipulating environmental big data with JASMIN, IEEE Big Data, https://doi.org/10.48550/arXiv.1204.3553, 2016.

Quilodrán-Casas, C., Silva, V., Arcucci, R., Heaney, C., Guo, Y. and Pain, C.: Digital twins based on bidirectional LSTM and GAN for modelling COVID-19, Neurocomputing, 470, 11-28, https://doi.org/10.48550/arXiv.2102.02664, 2022.

Gong, H., Cheng, S., Chen, Z. and Li, Q.: Data-Enabled Physics-Informed Machine Learning for Reduced-Order Modeling
490   Digital Twin: Application to Nuclear Reactor Physics, Nuclear Science and Engineering, 196, 668-693, https://doi.org/10.1080/00295639.2021.2014752, 2022a.

Bauer, P., Stevens, B. and Hazeleger, W.: A digital twin of Earth for the green transition, Nature Climate Change, 11, 80–83, https://doi.org/10.1038/s41558-021-00986-y, 2021.

Nadeem, K., Taylor, S.W., Woolford, D. and Dean, C.: Mesoscale spatiotemporal predictive models of daily human- and
495   lightning-caused wildland fire occurrence in British Columbia, International Journal of Wildland Fire, 29, 11, https://doi.org/10.1071/WF19058, 2020.

Mohan, A. and Gaitonde, D.: A Deep Learning based Approach to Reduced Order Modelling for Turbulent Flow Control using LSTM Neural Networks, https://doi.org/10.48550/arXiv.1812.04951, 2018.

Cheng, S., Prentice, I. C., Huang, Y., Jin, Y., Guo, Y. K. and Arcucci, R.: Data-driven surrogate model with latent data
500   assimilation: Application to wildfire forecasting}, Journal of Computational Physics, 464, 111302, https://doi.org/10.1016/j.jcp.2022.111302, 2022a.

Cheng, S., Chen, J., Anastasiou, C., Angeli, P., Matar, K. O. Guo, Y. K. Pain, C. C. and Arcucci, R.: Generalized Latent Assimilation in Heterogeneous Reduced Spaces with Machine Learning Surrogate Models, Journal of scientific computing, under review, https://doi.org/10.48550/arXiv.2204.03497, 2022b.

505   Gong, H., Yu, Y., Li, Q. and Quan, C.: An inverse-distance-based fitting term for 3D-Var data assimilation in nuclear core simulation, Annals of Nuclear Energy, 141, https://doi.org/10.1016/j.anucene.2020.107346, 2020.

Lorenc, C. A., Ballard, P. S., Bell, S. R., Ingleby, B. N., Andrews, F. L. P., Barker, D. M., Bray, R. J., Clayton, M. A., Dalby, T., Li, D., Payne, J. T. and Saunders, W. F.: The Met. Office global three-dimensional variational data assimilation scheme, Royal Meteorological Society, 126, 2991-3012, https://doi.org/10.1002/qj.49712657002, 2000.

510   Bonavita, M., Hólm, E., Isaksen, L. and Fisher, M.: The evolution of the ECMWF hybrid data assimilation system, Royal Meteorological Society, 142, 287-303, https://doi.org/10.1002/qj.2652, 2015

Ma, X., Lu, X., Yu, Y., Zhu, J. and Chen, J.: Progress on hybrid ensemble-variational data assimilation in numerical weather prediction, Journal of Tropical Meteorology, 20, 1188-1195, 2014.

Cheng, S., Argaud, J. P., Iooss, B., Lucor, D. aand Ponçot, A.: Error covariance tuning in variational data assimilation:
515   application to an operating hydrological model, Springer Nature, 35, 1019-1038, https://doi.org/10.1007/s00477-020-01933-7, 2020.

Gong, H., Cheng, S., Chen, Z., Li, Q., Quilodrán-Casas, C., Xiao, D. and Arcucci, R.: An efficient digital twin based on machine learning SVD autoencoder and generalised latent assimilation for nuclear reactor physics, Annals of Nuclear Energy, 179, 109431, https://doi.org/10.1016/j.anucene.2022.109431, 2022b.

520 Jain, P., Coogan, P. S., Subramanian, G. S., Crowley, M., Taylor, S. and Flannigan, D. M.: A review of machine learning applications in wildfire science and management, Environmental Reviews, 28, 478-505, https://doi.org/10.1139/er-2020-0019, 2020.

Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., et al.: Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review, 525 https://www.researchgate.net/publication/369380437_Machine_learning_with_data_assimilation_and_uncertainty_quantific ation_for_dynamical_systems_a_review, 2023.

Peyron, M., Fillion, A., Gürol, S., Marchais, V., Gratton, S., Boudier, P., Goret, G.: Latent space data assimilation by using deep learning, https://doi.org/10.48550/arXiv.2104.00430, 2021.

Amendola, M., Arcucci, R., Mottet, L., Casas, Q. C., Fan, S., Pain, C., Linden, P. and Guo, Y.: Data Assimilation in the Latent 530 Space of a Convolutional Autoencoder, ICCS 2021. Lecture Notes in Computer Science, 12746, https://doi.org/10.1007/978-3-030-77977-1_30, 2021.

Graves, A. and Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Networks, 18, 602-610, https://doi.org/10.1016/j.neunet.2005.06.042, 2005a.

Pechony, O. and Shindell, D. T.: Fire parameterization on a global scale, Journal of Geophysical Research, 114, 535 https://doi.org/10.1029/2009jd011927, 2009.

Rabin, S. S., Melton, J. R., Lasslop, G., Bachelet, D., Forrest, M., et al.: The Fire Modeling Intercomparison Project (FireMIP), phase 1: experimental and analytical protocols with detailed model descriptions, Geoscientific Model Development, 10, 1175–1197, https://doi.org/10.5194/gmd-10-1175-2017, 2017.

Bianchi, F. M., De Santis, E., Rizzi, A. and Sadeghian, A.: Short-Term Electric Load Forecasting Using Echo State Networks 540 and PCA Decomposition, IEEE, 3, 1931-1943, http://dx.doi.org/10.1109/ACCESS.2015.2485943, 2015.

Huang, Z., Xue, W., Mao, Q. and Zhan, Y.: Unsupervised domain adaptation for speech emotion recognition using PCANet, Multimedia Tools and Applications, 76, 6785-6799, http://dx.doi.org/10.1007/s11042-016-3354-x, 2017.

Masci, J., Meier, V., Ciregan, D., and Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction, International Conference on Artificial Neural Networks, http://dx.doi.org/10.1007/978-3-642-21735-7_7, 2011.

545 Graves, A. and Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM networks, Neural Networks, 4, 2047 – 2052, http://dx.doi.org/10.1109/IJCNN.2005.1556215, 2005b.

Vallino, J. J.: Improving marine ecosystem models: Use of data assimilation and mesocosm experiments, Journal of Marine Research - J MAR RES, 58, 117-164, http://dx.doi.org/10.1357/002224000321511223, 2000.

Lawless, A. S., Gratton, S. and Nichols, N. K.: Approximate iterative method for variational data assimilation, International 550 Journal for Numerical Methods in Fluids, 1, 1129-1135, https://doi.org/10.1002/fld.851, 2005.

Fulton, W.: Eigenvalues, invariant factors, highest weights, and Schubert calculus, Bulletin of The American Mathematical Society - BULL AMER MATH SOC, 37, 209-250. https://doi.org/10.1090/S0273-0979-00-00865-X, 2000.

Li, H., Li, Y., Wang, Z. and Li, Z.: Remaining Useful Life Prediction of Aero-Engine Based on PCA-LSTM, 63-66, https://doi.org/10.1109/CMMNO53328.2021.9467643, 2021.

555 Teckentrup, L., Harrison, S. P., Hantson, S., Heil, A., Melton, J. R., Forrest, M., Li, F., Yue, C., Arneth, A., Hickler, T., Sitch, S. and Lasslop, G.: Response of simulated burned area to historical changes in environmental and anthropogenic factors: a comparison of seven fire models, Biogeoscience, 16, 3883–3910. https://doi.org/10.5194/bg-16-3883-2019, 2019.