MDPI

Article

Parameter flexible wildfire prediction using machine learning techniques: forward and inverse modelling

Sibo Cheng*[1,2], Yufang Jin [3], Sandy P.Harrison[2,4], César Quilodrán-Casas [1],I. Colin Prentice[2,5], Yi-Ke Guo[1] and Rossella Arcucci[1,6]*

[1] Data Science Institute, Department of Computing, Imperial College London, UK
[2] Leverhulme Centre for Wildfires, Environment, and Society, London, UK
[3] Department of Land, Air and Water Resources, University of California, Davis, USA
[4] Georgina Mace Centre for the Living Planet, Department of Life Sciences, Imperial College London, UK
[5] Geography & Environmental Sciences, University of Reading, Reading, UK
[6] Department of Earth Science & Engineering, Imperial College London, UK
* Correspondence: sibo.cheng@imperial.ac.uk, r.arcucci@imperial.ac.uk

Abstract: Parameter identification for wildfire forecasting models often relies on case-by-case tuning or posterior diagnosis/analysis, which can be computationally expensive due to the complexity of the forward prediction model. In this paper, we introduce an efficient parameter flexible fire prediction algorithm based on machine learning and reduced order modelling techniques. Using a training dataset generated by physics-based fire simulations, the method forecasts burned area at different time steps with a low computational cost. We then address the bottleneck of efficient parameter estimation by developing a novel inverse approach relying on data assimilation techniques (latent assimilation) in the reduced order space. The forward and the inverse modellings are tested on two recent large wildfire events in California. Satellite observations are used to validate the forward prediction approach and identify the model parameters. By combining these forward and inverse approaches, the system manages to integrate real-time observations for parameter adjustment, leading to more accurate future predictions.

Keywords: Wildfire prediction; Machine learning; Reduced-order modelling; Convolutional autoencoder; Data Assimilation; Latent Assimilation; Parameter identification

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Remote Sens.* **2022**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Submitted to *Remote Sens.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

There has been a significant increase in wildfire frequency in many parts of the world in recent decades, causing losses of lives, huge economic costs and lethal effects of air pollution [1]. According to Verisks 2021 Risk Analysis [2], over four million properties in the U.S. were identified as being at high or extreme risk from wildfire. Accurate and efficient near-real time wildfire predictions, which can be used to make decisions about fire fighting strategy, are crucial for short-term fire emergency response as well as longer-term fire risk assessment [3]. However, the ability to forecast/simulate the spread of wildfires is currently limited because of the complexity of fire dynamics and the processes that control this [4]. State-of-the-art fire spread modelling techniques, based on, for instance, computational fluid dynamics (CFD) [5], statistical regression [6] or Cellular Automata (CA) [7,8], are not fast enough to produce real time predictions for massive wildfires. In recent years, data-driven and reduced-order modelling (ROM) techniques have been used to decrease the computational burden for high dimensional dynamical systems in several areas, including air pollution [9], fluid dynamics [10] and numerical weather prediction (NWP) [11]. Rather than learning the full physical space, the Machine Learning (ML) approaches seek to predict some latent variables of much lower dimension derived from data compression [10,12,13].

A variety of ROMs have been developed for dynamical systems, such as Principal Component Analysis (PCA) (also known as Proper Orthogonal Decomposition (POD) for dynamical systems) [14], information entropy based methods [15], Convolutional Autoencoder

(CAE) [16], Variational Autoencoder (VAE) [17] and the recently developed Singular Value Decomposition (SVD) Autoencoder (AE) [18,19]. The advantage of Deep Learning (DL)-based approaches (e.g., CAE) compared to projection-based methods (e.g., PCA) has been widely noted [10,20], especially for highly non-linear systems. In addition, non-intrusive ROM attempts to establish the input-output function of the model parameters and the reduced basis through interpolation [21–23], regression [24,25] or machine-learning [9,26] algorithms. A DL reduced surrogate modelling has been introduced in the recent work of [27] to perform near real-time fire forecasting for individual fire events. This surrogate model based on long short-term memory (LSTM) (a variant of Recurrent Neural Networks (RNNs)) was trained and evaluated in a latent space obtained via ROM. The efficiency and accuracy of the model were numerically demonstrated in reconstructing several massive fire events. However, although the RNNs provides accurate predictions in an iterative manner, individual surrogate models are required for different sets of model parameters (e.g., the impact coefficient of the wind speed, the empirical burning probability according to different vegetation types etc). As stated in [27] the generalisability of the approach in terms of model parameters and initial conditions should be improved in an operational context.

A second limitation of current fire modelling approaches is related to the difficulties of parameter estimation, especially when accounting for local features such as wind and fuel resources. As summarised in [8], a variety of algorithms [7,28,29] have been developed to nowcast/forecast regional fire progress relying on local geological and meteorological variables such as elevation, vegetation and wind speed. Parameter estimation is crucial for wildfire models to reduce the prediction bias. For both raster- (e.g., CA) and vector-based (e.g, level-set methods [30,31]) techniques, key model parameters are generally obtained via numerical experiments or by fire (or region) specific optimisation (e.g.,[7,32]). The requirement of parameter/hyperparameter tuning for different fire events (often "manually" as described in [33]) is not only computationally demanding but also limits generalisability. Much effort [33–37] has been made to enhance and generalise the parameter identification in wildfire modelling, which can be viewed as an inverse problem of fire forecasting. The work of [35] introduced a new uncertainty quantification approach for more efficient parameter calibration based on polynomial chaos expansion. [36] made use of an ensemble Kalman filter (KF) to assimilate parameters related to wind effects given observations of the actual burning area. However, despite achieving promising results in a small scale computational domain ($180m \times 180m$), the computational cost and sensitivity to observation error increased when the method was applied to large fire events since the KF is performed in the full observation space. In the very recent work of [37], the authors proposed a general algorithm scheme for parameter tuning in level set methods, which consists of iterative applications of the forward model between two consecutive observations of burned area until the stopping condition is satisfied. This iterative method can be computationally heavy. Moreover, as stated in [37], the approach can only perform parameter estimation between two real-time observations. In fact, parameter calibration/identification on ML surrogate models is even more challenging because the underlying physical parameters impact the model output in an indirect way where few explicit equations can be established [38].

In this paper, we develop a novel data-driven approach for burned area forecasting which is flexible regarding initial model parameters. Using inverse modelling based on Latent Assimilation (LA) [39,40] techniques, our approach can be used for parameter estimation with a low computational cost. We use a high-fidelity physics-based simulator, specifically an operating probabilistic CA simulator [7] with local geophysical features such as vegetation, slope and wind speed, to generate a dataset to train ML surrogate models for forward prediction. Our aim is to build a robust latent space which is valid for simulation data with different physical parameters. Thus, it will be more adaptable when compressing unseen observations. We therefore compare several different ROMs, namely PCA, CAE and SVD AE, for real wildfire events and evaluate their performance based on: i) the reconstruction accuracy with respect to unseen simulated data and satellite observations; ii) the forward fire prediction model once combined with ML approaches; iii) the inverse modelling for parameter identification; iv) the forecasting of future fire propagation using

estimated/adjusted parameters. After ROM, we aim to learn the dynamics of the latent variables which represent the evolution of the burned area. In this study, both shallow (K-nearest Neighbours (KNN), Random Forest (RF)) and deep machine learning algorithms (Multi Layer Percepton (MLP)) are implemented. The former has the advantage of less offline training complexity and good interoperability while deep learning methods often provide more accurate predictions. Physics-related parameters are considered as model inputs. Once the forward model is computed, online parameter tuning is crucial when predicting unseen fire events. Here, we employ the Generalised Latent Assimilation (GLA) approach [41] for parameter estimation by first applying the pre-trained data compression operators to the observed data to obtain latent vectors and then using these latent vectors as inputs in the Data Assimilation (DA) approach where the transformation operator (i.e.,the mapping from the state space to the observation space) is defined by the ML forward predictor. DA is a standard method for inverse problems when there is an initial estimate of the state vector or background state [42,43]. In this study, the state vector consists of a set of physical parameters. Our inverse modelling is capable of incorporating one or several observations at different time steps for parameter estimation, and thus the ML forward model can deliver more accurate future forecasting. Applying DA with a ML transformation function is challenging due to the complexity and the non-differentiability of ML functions. To tackle this bottleneck, we use the GLA approach introduced by [41]. Evaluating the ML function on a local ensemble of perturbed samplings around the background state, smooth surrogate functions are used to locally approximate the ML-based transformation operator, allowing DA to be carried out with a low computational cost. The GLA scheme is applied to parameter estimation and extended to a Four-dimensional variational data assimilation (4Dvar) [44] framework with time-dependent observations.

In summary:

- We propose a parameter flexible data-driven algorithm scheme for burned area forecasting which can combine different approaches of ROM and ML prediction techniques with a large range of model parameters.
- We develop an inverse model to address the bottleneck of parameter identification in fire prediction models by using the recently developed GLA algorithm,.
- We test the parameter flexible data-driven model and the inverse model for recent massive fire events in California where the data used for the assimilation are satellite observations[1] of burned area.

The resulting data-driven wildfire model with identified parameters allows fire propagation to be predicted in near-real time, which makes it suitable for use in the context of emergency response and fire fighting.

2. Data generation and study area

In this section we describe the models employed to generate the data used to train the parameter flexible data-driven model for burned area forecasting, and provide information on the study area and the satellite observations.

2.1. Cellular automata fire simulation

We make use of an operational CA model [7], for generating the training and testing data. The performance of this CA model was originally tested by predicting the Spetses fire in 1990 in Greece [7]. The model uses square meshes to simulate the random spatial spread of wildfires as shown in Figure 1. The use of regular square meshes reduces the computational cost for large fire events compared to using unstructured meshes. Four states are assigned to represent a cell at a discrete time,

- state 1: The cell can not be burned;
- state 2: The cell is burnable but has not been ignited;
- state 3: The cell is burning;

[1] ORNL DAAC. 2018. MODIS and VIIRS Land Products Global Subsetting and Visualization Tool. ORNL DAAC, Oak Ridge, Tennessee, USA

- state 4: The cell has been burned.

At each discrete time step, fire propagation into neighbour cells (i.e.,state 3 $\longrightarrow$ state 4) is stochastic, following the probability,

$$P_{\text{bun}} = p_h(1 + p_{\text{veg}})(1 + p_{\text{den}})p_s p_w \tag{1}$$

where p_h is the burning probability. p_{veg}, p_{den} , p_s and p_w are related to the local canopy density, canopy cover, landscape slope and wind speed/direction of the receiving cell, respectively. The actual values of these physical fields for corresponding areas were obtained from the Interagency Fuel Treatment Decision Support System [45]. The slope effect p_s is modelled following [46], that is,

$$p_s = \exp(a\theta_s). \tag{2}$$

where a is a dimensionless constant that can be adjusted. The slope angle θ_s in the CA modelling [7] is calculated by:

$$\theta_s = \begin{cases} \tan^{-1}\left(\frac{E_1 - E_2}{l}\right) & \text{for adjacent cells} \\ \\ \tan^{-1}\left(\frac{E_1 - E_2}{\sqrt{2}l}\right) & \text{for diagonal cells} \end{cases} \tag{3}$$

where E_1 and E_2 are the altitude of the cells and l is the cell length. To integrate the wind effect, we adapt the modelling in [7], that is,

$$p_w = \exp(c_1 V_w) f_t, \quad f_t = \exp(V_w c_2 (cos(\theta_w) - 1)) \tag{4}$$

where V_w denotes the wind speed in m/s and θ_w represents the angle between the wind direction and the potential fire spread direction as illustrated in Figure 1. c_1 and c_2 are two tunable coefficients of wind effect. Spotting, i.e. fire starts beyond the fire edge due to wind-transport of flaming embers, is also considered [7]. The wind data of the corresponding area and date for different fire events were extracted from the dataset of [47]. Wind speed and direction are considered as spatially constant over the $27km \times 27km$ grid cell, as shown in Figure 1. The time step of the CA simulation is roughly equivalent to 30 minutes. To be exact, 40 CA steps are equivalent to 24 hours in real time [27].

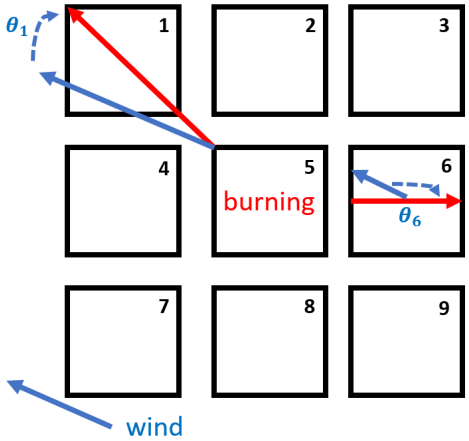


Figure 1. Wind effect in the CA modelling

The operational parameters p_h, a, c_1, c_2 can have a large impact on the fire forecast. In [7], these parameters are fixed as

$$p_h = 0.58, \quad a = 0.078, \quad c_1 = 0.045, \quad c_2 = 0.131, \tag{5}$$

Table 1. Study areas of the Chimney and the Ferguson wildfire events in this work. The latitude and the longtitude represent the centre of the fires. The averaged wind speed of the first 6 days of fire propagation is also indicated.

Fire	latitude	longitude	area	duration	start	wind
Chimney	37.6230	-119.8247	$\approx 246\text{km}^2$	23 days	13/8/2016	23.56mph
Ferguson	35.7386	-121.0743	$\approx 185\text{km}^2$	36 dyas	13/7/2018	18.54mph

resulting from the minimisation of a cost function that fits the observations in the specific fire event modelled. These values are used as initial guess in the parameter identification algorithm.

2.2. Study area and observation data

We evaluate the performance of our modelling approaches using two recent massive fire events in California, namely the Chimney fire in 2016, and the Ferguson fire at 2018 (Table 1). The vegetation density in the area of the Chimney fire was higher than that in the Ferguson fire, resulting in a much faster rate of propagation. Thus, the two fires provide contrasting behaviour and thus a good basis for assessing the robustness of our modelling. The area indicated in Table 1 represents the study area of both fire events, which is different from the total burned area. The averaged wind speed (mph, meters per hour) for the first 6 days is also indicated for both fire events. We use active fire data generated from Moderate Resolution Imaging Spectroradiometer (MODIS) and Visible Infrared Imaging Radiometer Suite (VIIRS) satellites. MODIS provides thermal observations globally four times a day (Terra at 10:30 and 22:30; Aqua at 13:30 and 01:30 local time) at a resolution of about 1km ([48]). VIIRS thermal data provides improved fire detection capabilities every 12 hours (at 13:30 and 1:30 local time) [49]. In this study, the level 2 VIIRS I-band active fire product (VNP14IMG) with a resolution of 375 m is combined with the MODIS fire products (MOD14 and MYD14) at 1km to derive continuous daily fire perimeters, using the natural neighbour geospatial interpolation method [50]. Both MODIS and VIIRS data are available 2.5 hours after the observations time, allowing near real-time fire updating.

3. A parameter flexible data-driven model for burned area forecasting: Methodology

In this section we describe the models we designed based on existing algorithms for developing a parameter flexible data-driven model for burned area forecasting, specifically the models for compressing/reducing the data, the predictive models in the low-dimensional space and the LA approach for parameter estimation.

3.1. Reduced-order modelling

The CA model provides a source of time series data which is used to train the ML predictive models. The size of the final dataset mandates the use of ROM techniques. To build a low dimensional latent space for burned area with high accuracy of reconstruction after decoding/decompression, we integrate and compare three different compression approaches. The burned area in the full space (obtained from simulators or satellite observations) and the encoded low-dimensional space at time t is denoted as $\mathbf{y}_t$ and $\tilde{\mathbf{y}}_t$ respectively.

3.1.1. Principle component analysis

We first apply the PCA method for encoding the physical fields of burned areas. A set of n_y simulated fields $\{\mathbf{y}_t^{(i)}\}_{\{i=0\dots n_y\}}$ at a fixed time t are flattened and combined in a matrix,

$$\mathbf{Y}_t = \left[\mathbf{y}_t^{(0)}\middle|\mathbf{y}_t^{(1)}\middle|\dots\middle|\mathbf{y}_t^{(n_y)}\right]. \tag{6}$$

In this study, n_y represents the number of different sets of model parameters, which will generate different burned areas at time t through the CA modelling. The ROMs at different time steps are performed individually. For the sake of simplicity, we adopt the notation $\mathbf{y}, \mathbf{Y}$ for $\mathbf{y}_t, \mathbf{Y}_t$ in the rest of Section 3.1. We compute and decompose the empirical covariance $\mathbf{C_y}$,

$$\mathbf{C_y} = \frac{1}{n_\mathbf{y} - 1}\mathbf{Y}\mathbf{Y}^T = \mathbf{L_Y}\mathbf{D_Y}{\mathbf{L_Y}}^T. \tag{7}$$

Here each column of $\mathbf{L_Y}$ is a principal component (PC) of $\mathbf{C_y}$ and $\mathbf{D_Y}$ is a diagonal matrix formed by the associated eigenvalues $\{\lambda_{\mathbf{Y},i}, i = 0, ..., n_y - 1\}$ in decreasing order, i.e.,

$$\mathbf{D_Y} = \begin{bmatrix} \lambda_{\mathbf{Y},0} & & \\ & \ddots & \\ & & \lambda_{\mathbf{Y},n_y-1} \end{bmatrix}. \tag{8}$$

A projection matrix $\mathbf{L}_{\mathbf{Y},q}$ of truncation parameter $q(1 \leq q \leq n_y)$ is constructed by extracting the q first columns in $\mathbf{L_Y}$. For each flattened field $\mathbf{y}$, the PCA-encoded latent vector $\tilde{\mathbf{y}}$ is obtained via

$$\tilde{\mathbf{y}} = {\mathbf{L}_{\mathbf{Y},q}}^T\mathbf{y}. \tag{9}$$

An approximation of the full field vector can be computed through PCA-decoding,

$$\mathbf{y}_{\text{PCA}} = \mathbf{L}_{\mathbf{Y},q}\tilde{\mathbf{y}} = \mathbf{L}_{\mathbf{Y},q}{\mathbf{L}_{\mathbf{Y},q}}^T\mathbf{y}. \tag{10}$$

3.1.2. Convolutional autoencoding

Autoencoding is an unsupervised DL approach that learns how to perform efficient data compression from the training dataset. A typical autoencoder consists of an encoder $\mathcal{E}$ which compresses the input vector to the latent variables and a decoder $\mathcal{D}$ to reconstruct the data back from the low-dimensional representation,i.e.,

$$\tilde{\mathbf{y}} = \mathcal{E}(\mathbf{y}) \quad \text{and} \quad \mathbf{y}_{\text{CAE}} = \mathcal{D}(\tilde{\mathbf{y}}). \tag{11}$$

Both encoders and decoders can be constructed via Neural Networks (NNs), but this is problematic for high-dimensional data because of the large number of parameters. Furthermore, NNs treat all input variable similarly causing difficulties in representing local image features. CAE overcomes these problems by using convolutional layers in the NN structure. Convolutional layers make use of multi-dimensional filters to capture local patterns. The size of the filter is fixed in each convolutional layer. Pooling layers are then added to reduce the dimension (i.e., number of neurons). When decoding, Upsampling layers can be used to reconstruct the NN input. For more details about Convolutional Neural Network (CNN) and CAE, interested readers are referred to [51]. CAE requires training of fewer parameters and recognizes spatial patterns thanks to multi-dimensional filters. We train the convolutional encoder and decoder jointly using the mean square error (MSE) loss function,

$$J(\mathcal{E}, \mathcal{D}) = \frac{1}{N_{\text{train}}}\sum_{j=1}^{N_{\text{train}}} ||\mathbf{y}_j - \mathcal{D} \circ \mathcal{E}(\mathbf{y}_j)||^2. \tag{12}$$

3.1.3. Singular value decomposition autoencoding

Since training a CAE is time consuming for high dimensional problems, we implement a training-efficient autoencoder, known as SVD AE or POD AE, which combines the PCA approach and the ML-based AE. Dimension reduction is performed in two steps. PCA is

first applied to obtain the full set of PCs of the dynamical system, followed by a dense autoencoder $(\mathcal{E}', \mathcal{D}')$ with fully connected layers to reduce the system dimensions further,

$$\tilde{\mathbf{y}} = \mathcal{E}'\big(\mathbf{L_Y}^T \mathbf{y}\big), \quad \text{while} \quad \mathbf{y}_{\text{SVD AE}} = \mathbf{L_Y}\mathcal{D}'(\tilde{\mathbf{y}}). \tag{13}$$

Thus, both the input and the ouput of the encoder $\mathcal{E}'$ and the decoder $\mathcal{D}'$ are the principle components of the full system. Combining PCA and ML autoencoding means this approach has both the efficiency of PCA and the power of CAE for dealing with chaotic systems. SVD AE can handle data with both structured or unstructured geometry.

3.2. Forward problem: machine learning prediction

We implement different ML regression approaches to predict the latent variables $\tilde{\mathbf{y}}_t$ at different time steps using model parameters $\mathbf{x} = [c_1, c_2, p_h, a]$, i.e.,

$$\tilde{\mathbf{y}}_t = f_t^{\text{ML}}(\mathbf{x}). \tag{14}$$

Training is performed on an ensemble of perturbed parameter sets, generated via Latin Hypercube Samplings (LHSs) within the range,

$$p_h \in [0.00, 0.70], \quad a \in [0.00, 0.14], \quad c_1 \in [0.00, 0.12], \quad c_2 \in [0.00, 0.40], \tag{15}$$

based on the parameter values given in the literature and some preliminary experimentation. We generated a 2300-member ensemble (i.e., 2300 simulations with different model parameters), which was then split into training and testing datasets of 1000 members and a 300-member validation dataset. The training and the testing datasets are used for both fire events, and the two fire events were simulated via CA until 8 days after the ignition.

3.2.1. Random forest regression

We first use random decision forests RF model, an ensemble ML approach based on Decision Trees (DTs), to predict the latent variables derived from ROM. RFs employ a *bagging* approach, which trains a set of individual prediction models simultaneously using a random subset of the training data. The number of individual models n_{DT}, the maximum depth d_{DT} of each DT and the number of features n_{features} considered for the tree split are hyperparameters in RFs modelling and obtained based on hyperparameter tuning on the validation dataset. Here, we use Classification And Regression Tree (CART) with *gini* coefficients [52] for tree splitting in the DT framework.

3.2.2. K-nearest neighbours regression

We also use KNN for latent variable prediction. KNN is a non-parametric machine learning method where the model outputs only depend on the k local training samples (denoted as $N_k(\mathbf{x})$) which are the closest to the input vector $\mathbf{x}$. As a consequence, the function is only approximated by local samples and no pre-training is required. The weight of each training sample $\mathbf{x}_i$ (for $\mathbf{x}_i \in N_k(\mathbf{x})$) is proportional to the inverse of the $\mathcal{L}_2$ distance, i.e.,

$$\mathbf{y} = \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} w_i \mathbf{y}_i, \quad \text{where} \quad w_i = \frac{\frac{1}{||\mathbf{x}-\mathbf{x}_i||_2}}{\sum_{\mathbf{x}_j \in N_k(\mathbf{x})} \frac{1}{||\mathbf{x}-\mathbf{x}_j||_2}}. \tag{16}$$

To search the ensemble of nearest samples (i.e.,$N_k(\mathbf{x})$), we use the $kd-tree$ algorithm [53] which is appropriate for low-dimensional data (here, $\dim(\mathbf{x}) = 4$). The maximum leaf size of the $kd-tree$ is fixed as 30.

3.2.3. Multi layer perceptron

We also apply a deep learning (MLP) approach in the latent space to predict the fire propagation. Since the problem is of a relatively small dimension (i.e., $\dim(\mathbf{x}) = 4$ and $\dim(\mathbf{y}) \leq 50$), a fully-connected NN structure with two hidden layers is employed, that is,

$$\dim(\mathbf{x}) = 4 \longrightarrow n_{\text{MLP},1} \longrightarrow n_{\text{MLP},2} \longrightarrow \dim(\mathbf{y}), \tag{17}$$

where $n_{\text{MLP},1}$ and $n_{\text{MLP},2}$ denote the number of neurons in both hidden layers, respectively. The pipeline of the forward modelling, involving data generating, ROM, and prediction in the latent space is illustrated in Figure 2.

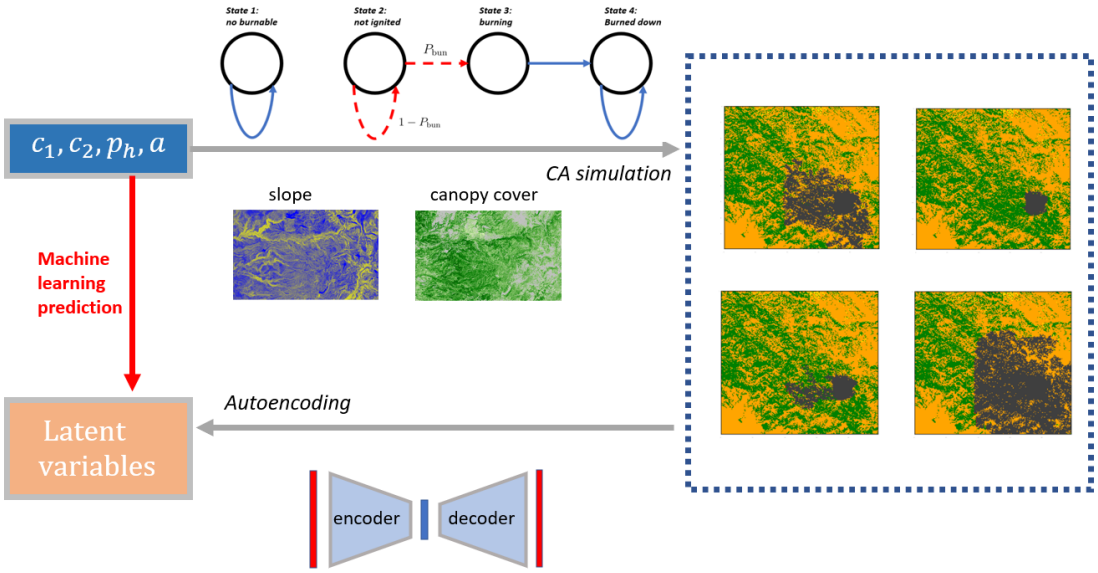


Figure 2. Flowchart of the forward prediction model with CA and ROM for a specific ecoregion. The data generation is performed with a set of parameters perturbed using LHS

3.3. Inverse problem: latent data assimilation

The choice of model parameters significantly impacts the fire spread forecast. Thus, parameter calibration using real-time observations is crucial for both physics- and ML-based prediction models. We develop a novel parameter identification method based on the GLA algorithm with time-variant observations to solve the inverse problem. To reduce the computational burden, the observations are first encoded into low-dimensional latent spaces.

3.3.1. Four dimensional variational approach

We employ data assimilation for model parameter identification by fitting the encoded observations $\{\tilde{\mathbf{y}}_t\}$ within an observing window $T_{\text{obs}} = \{t_1, ..., t_{n_{obs}}\}$ by implementing 4Dvar algorithm [44] using the ML forward models and defining the loss function as:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{x}_b||^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{t\in\mathbf{T}_{\text{obs}}} ||\tilde{\mathbf{y}}_t - f_t^{\text{ML}}(\mathbf{x})||^2_{\tilde{\mathbf{R}}_t{}^{-1}} \tag{18}$$

where $\mathbf{B}, \tilde{\mathbf{R}}_t$ represent the prior error covariance matrices [54,55] associated to the states and encoded observations, respectively. These covariance matrices are set to be diagonal in the latent space in this study. Minimising the 4Dvar loss function (equation (18)) leads to the analysis state,

$$\mathbf{x}_a = \underset{\mathbf{x}}{\text{argmin}}\Big(\mathcal{J}(\mathbf{x})\Big). \tag{19}$$

The minimisation of 4Dvar is performed using the LBFGS approach [56], where each minimisation iteration can be written as:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - l_{\text{rate}}\left[\text{Hessian}(\mathcal{J})\mathbf{x}_k\right]^{-1}\nabla\mathcal{J}(\mathbf{x}^{(k)}). \tag{20}$$

Here k is the current iteration number and $l_{\text{rate}} > 0$ is the learning rate of the descent algorithm.

$$\text{Hessian}\Big(\mathcal{J}(\mathbf{x} = [x_0, ..., x_{n_x-1}])\Big)_{i,j} = \frac{\partial^2\mathcal{J}}{\partial x_i \partial x_j} \tag{21}$$

is the Hessian matrix of the cost function $\mathcal{J}$. The computation of Hessian($\mathcal{J}$) is thus required for implementing variational assimilation.

3.3.2. Generalised Latent Assimilation

Since the model parameters and the encoded observations $\tilde{\mathbf{y}}_t$ are mapped through ML functions, the direct computation of Hessian($\mathcal{J}$) can be difficult due to the high non-linearity and the large number of model parameters. The recent work of [41] proposed a new assimilation approach GLA to tackle this bottleneck in current Latent Assimilation methods, where local Polynomial Regressions (PRs) are performed in a neighbourhood of the background state to build a smooth local surrogate function to facilitate the computation of the Hessian matrix in the optimisation loops. Here, LHS is performed to build a PR learning ensemble $\{\mathbf{x}_b^q\}_{q=1..n_s}$ around the background state within certain range r_s (i.e., sampling from $[(1 \pm r_s)x_0, ..., (1 \pm r_s)x_{n_x-1}]$). We then fit the ML model output by a local polynomial function $\tilde{\mathcal{H}}_t^p$, i.e.,

$$\tilde{\mathcal{H}}_t^p = \underset{\mathcal{P}\in P(d_p)}{\text{argmin}}\Big(\sum_{q=1}^{n_s} ||p(\mathbf{x}_b^q) - f_t^{\text{ML}}(\mathbf{x}_b^q)||_2^2\Big)^{1/2}, \tag{22}$$

where $P(d_p)$ represents the set of polynomial functions of degree d_p. $\{d_p, n_s, r_s\}$ are considered as hyperparamters of the algorithm. We extend the GLA to a 4Dvar framework with time-dependent observations, as shown in Algorithm 1.

Algorithm 1: 4Dvar GLA

Inputs: $\mathbf{x}_b, \mathbf{B}, \{\tilde{\mathbf{R}}_t\}, \{\tilde{\mathbf{y}}_t\}, \{f_t^{\text{ML}}\}$ for $t \in \mathbf{T}_{\text{obs}}$
Paramters: d_p, r_s, n_s
$\mathbf{x}_0 = \mathbf{x}_b$, $k = 0$

for $t \in \boldsymbol{T}_{obs}$ **do**
  $\{\mathbf{x}_b^q\}_{q=1..n_s} = \text{LHS Sampling}_{\{d^p, r_s, n_s\}}(\mathbf{x}_b)$
  for $q = 0$ *to* n_s **do**
    $\tilde{\mathbf{y}}_t^q = f_t^{\text{ML}}(\mathbf{x}_b^q)$
  end
  $\tilde{\mathcal{H}}_t^p = \text{PR}_{\text{train}}\big(\text{input:}\{\mathbf{x}_b^q\}, \text{output:}\{\tilde{\mathbf{y}}_t^q\}, q = 1..n_s\big)$
end
$\mathbf{x}_a = \underset{\mathbf{x}}{\text{argmin}}\Big(\frac{1}{2}||\mathbf{x} - \mathbf{x}_b||_{\mathbf{B}^{-1}}^2 + \frac{1}{2}\sum_{t\in\mathbf{T}_{\text{obs}}} ||\tilde{\mathbf{y}}_t - \tilde{\mathcal{H}}_t^p(\mathbf{x})||_{\tilde{\mathbf{R}}_t{}^{-1}}^2\Big)$
outputs: $\mathbf{x}_a$

Once the analysis state (i.e., assimilated parameters) $\mathbf{x}_a$ is obtained, forecasts of future burned area at time t can be made,

$$\mathbf{y}_t^{\text{pred}} = D_y\big(f_t^{\text{ML}}(\mathbf{x}_a)\big). \tag{23}$$

The pipeline of the inverse modelling for parameter estimation is illustrated in Figure 3.

3.4. Hyperparameter tuning

To select the most appropriate data compression methods, we apply the three different ROM approaches and evaluate their performance on the unseen test dataset and satellite observations. Forward models are trained on simulated burned areas individually 2, 4, 6 days after the ignition of each fire in order to capture the time of the fastest propagation. We set the burned and burning cells to 1 while unburned cells are set to 0. A threshold of 0.5 is set to classify the output cells after reconstruction to the full space. The performance of different ROMs and prediction models is evaluated using both the test dataset (obtained via CA simulations) and the satellite observations. Hyperparameter tuning is carried out for both the forward and inverse models on the validation dataset as shown in Table 2. For both fire events, the NN structures of CAE and SVD AE are similar. The exact networks for the Chimney fire are shown as an example in Table 3.

Table 2. Hyperparameter grid search space

Model/Hyperparameters	Grid search space	Final set
CAE		
Filter, Strides, Pooling size	/	Table 3
Activation	{ReLu, LeakyReLu, Sigmoid}	Table 3
Optimizer	{Adam, SGD}	Adam
Batch size	$\{16, 32, 64\}$	32
RF		
split criteria	{ 'gini', 'entropy'}	'gini'
n_{DT}	$\{10, 50, 100\}$	100
n_{features}	{ 'log2','sqrt'}	'sqrt'
KNN		
k	{ 5,10,20}	5
Metric	$\{L_1, L_2\}$	L_2
MLP		
$n_{\text{MLP},1}$	{ 10,20,30}	20
$n_{\text{MLP},2}$	{ 30,40,50}	30
Activation	{ReLu, LeakyReLu, Sigmoid}	
Optimizer	{Adam, SGD}	Adam
GLA		
d_p	2-6	4
n_s	$\{200, 500, 1000, 2000\}$	1000
r_s	$50\% - 200\%$	80%

Table 3. NN structure of the CAE with ordered meshes where the latent dimension $q \in \{10, 20, 30, 40, 50\}$.

Layer (type)	Output Shape	Activation
Encoder		
Input	$(899, 982, 1)$	
Conv 2D (10×10)	$(899, 982, 4)$	ReLu
MaxPooling 2D (5×5)	$(180, 197, 4)$	
Conv 2D (4×4)	$(180, 197, 4)$	ReLu
MaxPooling 2D (3×3)	$(60, 66, 8)$	
Conv 2D (3×3)	$(60, 66, 8)$	ReLu
MaxPooling 2D (3×3)	$(20, 22, 8)$	
Conv 2D (2×2)	$(20, 22, 8)$	ReLu
MaxPooling 2D (2×2)	$(10, 11, 8)$	
Flatten	880	
Dense (q)	q	LeakyReLu (0.3)
Decoder		
Input	q	
Dense (110)	110	LeakyReLu (0.3)
Reshape	$(10, 11, 1)$	
Conv 2D (2×2)	$(10, 11, 8)$	ReLu
Upsampling 2D (2×2)	$(20, 22, 8)$	
Conv 2D (3×3)	$(10, 11, 8)$	ReLu
Upsampling 2D (3×3)	$(60, 66, 8)$	
Conv 2D (4×4)	$(60, 66, 8)$	ReLu
Upsampling 2D (3×3)	$(180, 198, 8)$	
Conv 2D (5×5)	$(180, 198, 4)$	ReLu
Upsampling 2D (5×5)	$(900, 990, 4)$	
Cropping 2D $(1, 8)$	$(899, 982, 4)$	
Conv 2D (8×8)	$(899, 982, 1)$	Sigmoid

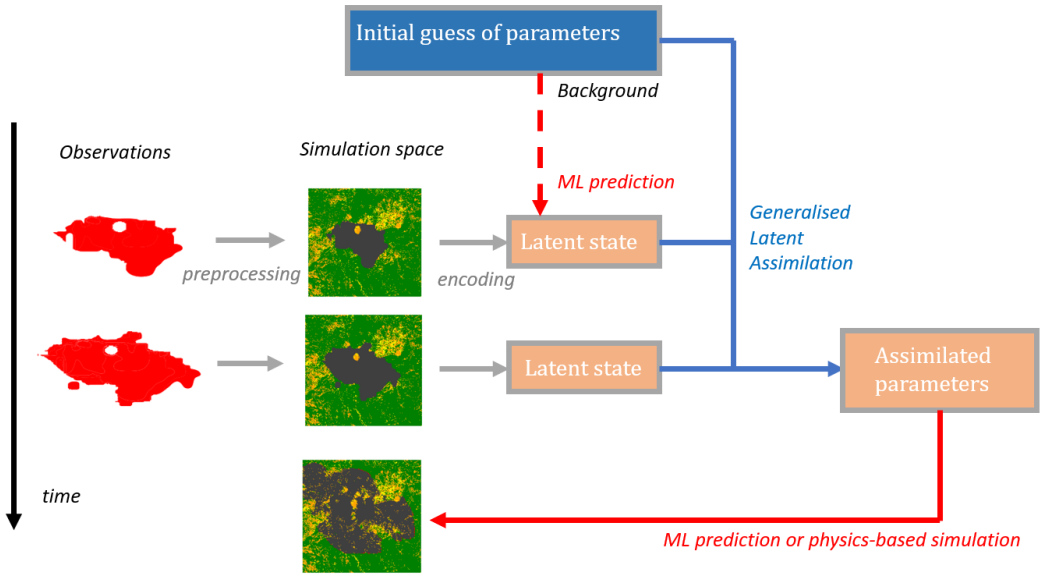


Figure 3. Flowchart of the inverse model for parameter identification using GLA

4. Results and analysis

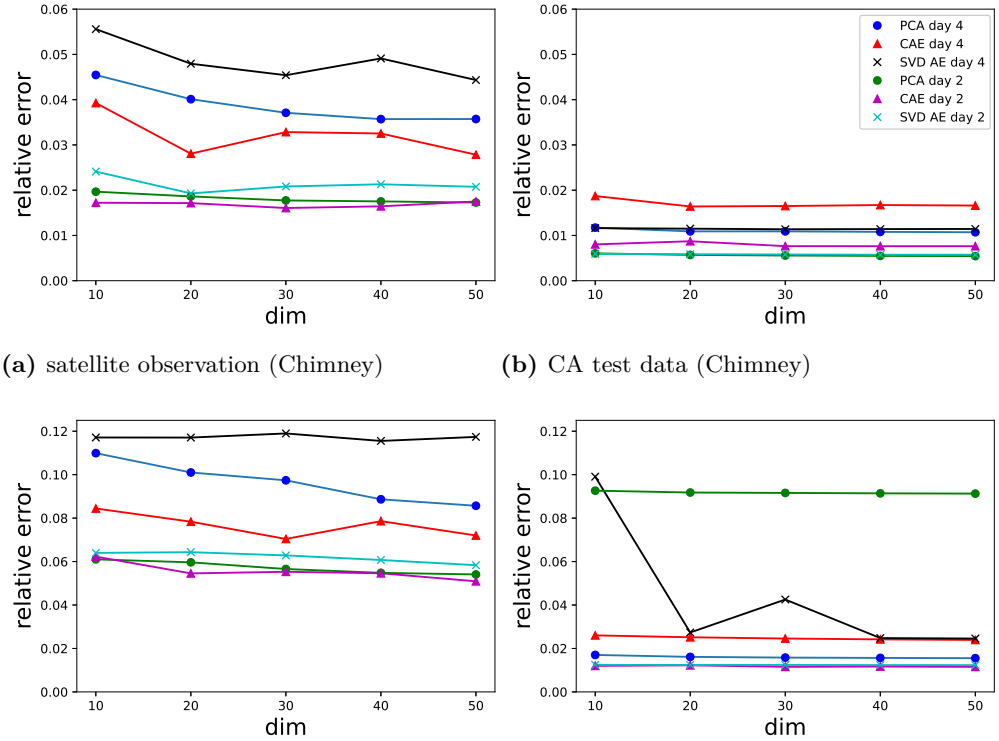
In this section, we present and analyse the numerical results from ROM, ML forecasting and GLA parameter estimation using simulated data and satellite observations.

4.1. Reconstruction accuracy of reduced order modellings

To compare different ROMs, we compute the relative reconstruction error after decoding,

$$\epsilon_{\text{rec}} = \frac{||\mathbf{y} - \mathbf{y}_{\{\text{PCA, CAE, SVD AE}\}}||_1}{\dim(\mathbf{y})}, \tag{24}$$

where $||.||_1$ denotes the one-norm, in other words, the number of mis-predicted pixels. With the same number of latent variables for different ROMs, ϵ_{rec} represents the efficiency of data compression strategies which is crucial for predictions latter. The evolution of ϵ_{rec} against the truncation parameter (i.e., the dimension of the latent space) is shown in Figure 4 for both the Chimney and the Ferguson fire events at both the day 2 and the day 4. The estimation of ϵ_{rec} was made using all 1000 simulations in the test dataset of CA simulations (Figures 4(a,c)) while the ϵ_{rec} for the real fire (Figures 4(b,d)) is evaluated using the satellite observation (a single image) at day 4.



(a) satellite observation (Chimney) **(b)** CA test data (Chimney)

(c) satellite observation (Ferguson) **(d)** CA test data (Ferguson)

Figure 4. Relative reconstruction error of satellite observations and CA simulations in the test dataset against the dimension of the latent space

Reconstruction errors increase with time from the start of the fire for both fires as shown in Figure 4 (b, d). The reconstruction errors for the satellite images are larger than those for the test data, which is to be expected since the autoencoders were trained entirely using simulated data. The PCA and SVD AE approaches have lower reconstruction errors than CAE in the test dataset generated by CA simulations. However, CAE shows better performance against the satellite data than either PCA or SVD AE. The differences between the three approaches are also seen in the predictions of burnt area. In Figures 5 and 6, we show the reconstruction of different ROMs for two sets of model parameters where the original CA simulations are represented by sub-figures (a,i) for each fire. The PCA and SVD

AE approaches are better at representing local variability in burnt area in the test dataset (Figures 5 and 6 f-h, n-p). However, overfitting to the CA dataset means that they are less good than CAE at predicting the satellite observations (Figure 7). On the other hand, CAE produces more continuous burned areas compared to PCA and SVD AE because of the convolutional layers in compressing and reconstructing the images. These findings are consistent with the results obtained in Figure 4 and the earlier analysis of other fire events [27] . The Ferguson fire has a higher average vegetation density, leading to a considerably faster fire spread than in the Chimney fire (Figure 5 and 6). This is seen in both the CA simulations and the observations of burned area. It would be possible to improve the performance of these ML-based ROMs, particularly the CAE and SVD AE) approaches, by increasing the size of the training dataset, but this would increase the computational cost since the CA simulation is relatively slow. Because most of the ROMs achieve a stable performance for $q \geq 30$, we fix the dimension of the latent space as $q = 30$ for both forward and inverse modellings in the rest of this paper.

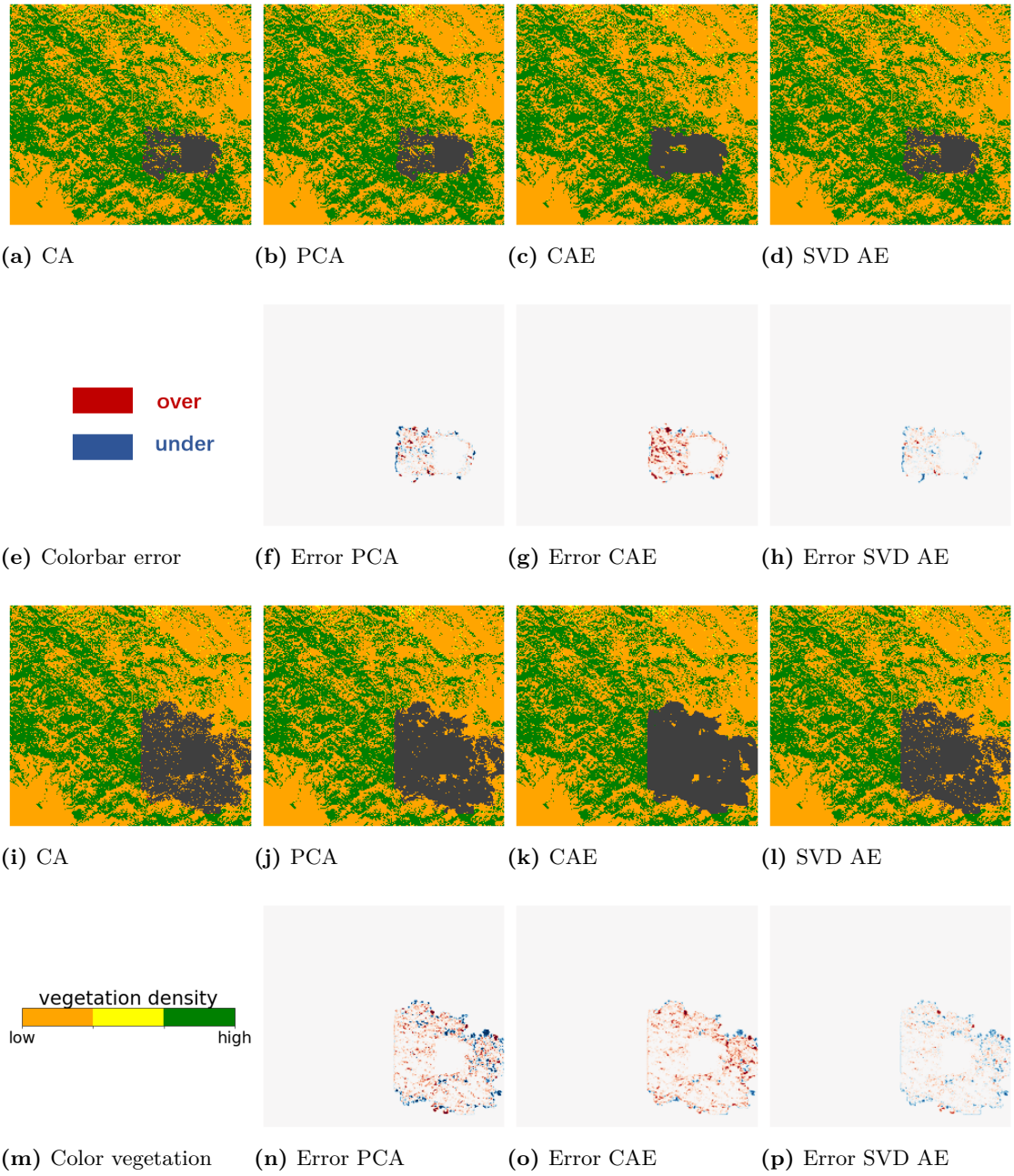
4.2. Prediction performance of the forward model

Having compressed the full space of burned area into low-dimensional latent variables, we apply the models introduced in Section 3.2 to predict fire propagation in these reduced spaces using the same training, validation and test datasets as for ROMs. Comparisons of the predictions of latent variables on the test dataset against the true compressed values is illustrated in Figure 8. We focus on the most important PC (i.e., the one associated to $\lambda_{\mathbf{Y},0}$) for the PCA approach and the latent variables with highest variance for CAE and SVD AE. For clarity, the samples in each sub-figure are reordered in increasing order along the true compressed values (i.e., red curves). Since CA generates probabilistic simulations, the same set of model parameters may lead to different simulations of burned area which introduces some uncertainties in the ML approaches as observed in Figure 8. Nevertheless, the predictions for the test dataset are still reasonably accurate and show the advantage of DL-based approaches. We show the forecasting error ϵ_{pred} (defined similarly to ϵ_{rec}) in the full space after decoding in Table 4. Good prediction accuracy (with an error below 5% in the full space) can be achieved by each combination of ROM and ML approaches. The PCA and SVD AE approaches produce slightly more accurate predictions compared to CAE-based approaches due to the reconstruction error on the CA test dataset, as discussed in Section 4.1. The predicted burned areas using CAE are more continuous.

The training time of ML approaches (i.e., KNN, RF and MLP) in the latent space differs slightly when combined with different ROMs. As an example, we show the training time in the reduced space of CAE in Table 5. The main computational cost for online prediction is decoding, since the evaluation of trained ML algorithms is very fast. We show the results of RF coupling with different ROM approaches, but using KNN or MLP produces similar results. In Table 5, only the training of CAE is performed with one NVIDIA P100 GPU (RAM of 16 Gb) of Google Colab environment. The other works including CA, are carried out with Colab Intel CPUs (2.30GHz and 26.75Gb RAM). The ROM and ML-based online predictions (especially with CAE and PCA), are much faster than the physics-based CA simulations. The online prediction of SVD AE is relatively slow because its decoding involves a large number of principle components. Comparing the results in Tables 4, 5 and Figure 4, SVD AE is out-performed by PCA in terms of both reconstruction/prediction accuracy and online efficiency. We, therefore, only focus on PCA and CAE for parameter estimation.

Table 4. Averaged prediction error for Chimney and Ferguson fires at day 4 in the full physical space after decoding. The dimension of the latent space is fixed as $q = 30$.

ML approache	Chimney			Ferguson		
	PCA	CAE	SVD AE	PCA	CAE	SVD AE
KNN	1.87%	4.95%	1.88%	2.00 %	3.30%	2.73%
RF	1.70%	4.91%	1.80%	1.89%	3.27%	2.45%
MLP	1.71%	4.54%	1.59%	2.13%	3.11%	2.54%



(a) CA **(b)** PCA **(c)** CAE **(d)** SVD AE

(e) Colorbar error **(f)** Error PCA **(g)** Error CAE **(h)** Error SVD AE

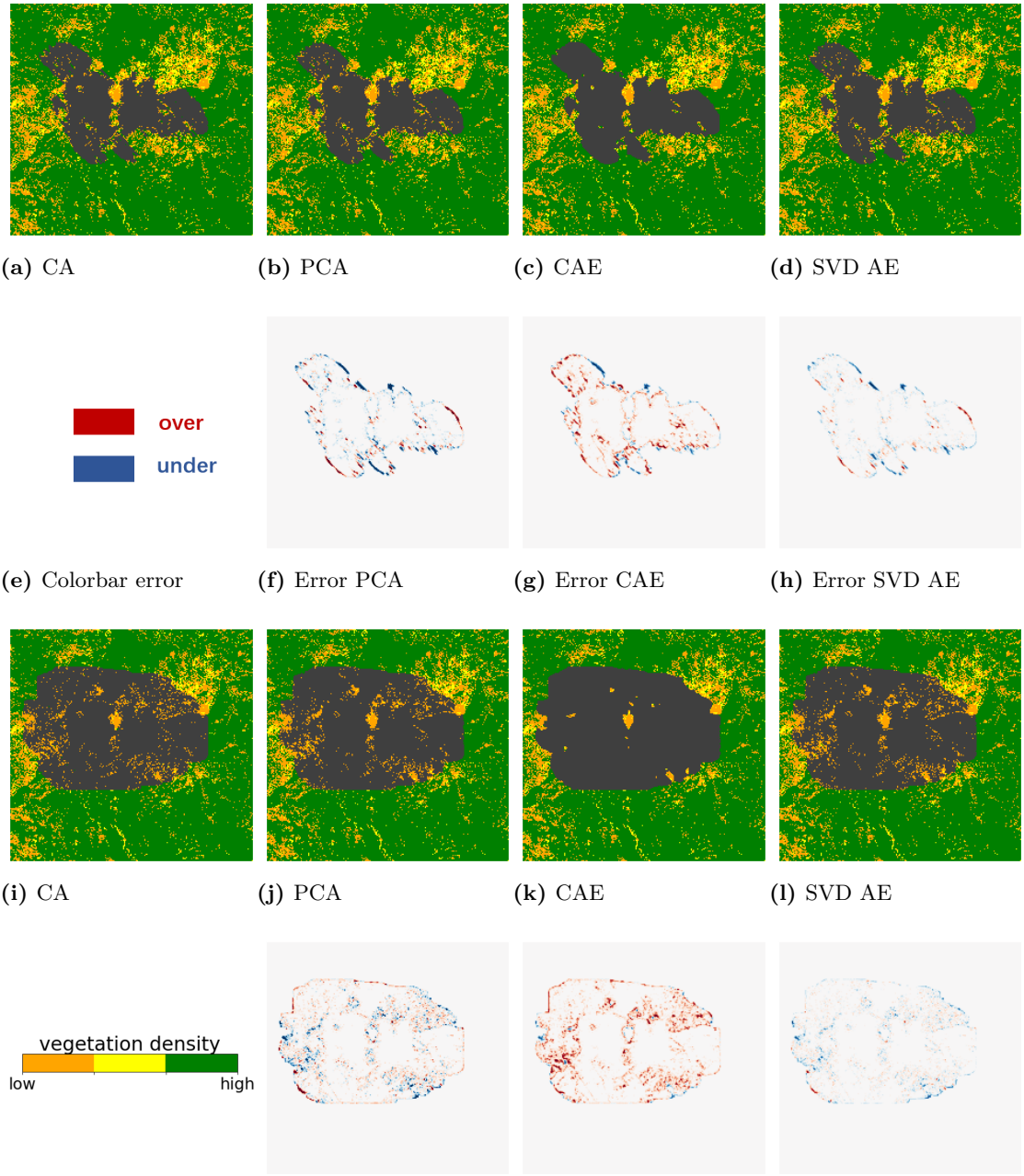
(i) CA **(j)** PCA **(k)** CAE **(l)** SVD AE

(m) Color vegetation **(n)** Error PCA **(o)** Error CAE **(p)** Error SVD AE

Figure 5. Reconstruction of CA simulations for 2 examples in the test dataset of the Chimney fire at day 4

4.3. Parameter estimation of the inverse model

We use observations of day 2 and day 4 to improve parameter estimation following the flowchart shown in Figure 3 and test the algorithm performance for the accuracy of predictions on day 6 for both fire events. The CA simulation using the true parameters in the test dataset is considered as the ground truth. Only slight differences are found between the third and the fourth columns of Figure 9, which confirms the accuracy of the forward modelling. The predictions with assimilated parameters (i.e., CAE-MLP($\mathbf{x}_a$)) are much closer to the CA simulations (i.e., CA($\mathbf{x}_t$)) regarding the background predictions (i.e., CAE-MLP($\mathbf{x}_b$)) using by-default values (equation (5)) of parameters. The strength of GLA when dealing with satellite data can be seen in Figure 10, where the preprocessed satellite images of day 2 and day 4 (Figure 7 (a,i)) have been used as observations in GLA for parameter estimation after encoding. Since both the forward prediction models and the



(a) CA **(b)** PCA **(c)** CAE **(d)** SVD AE

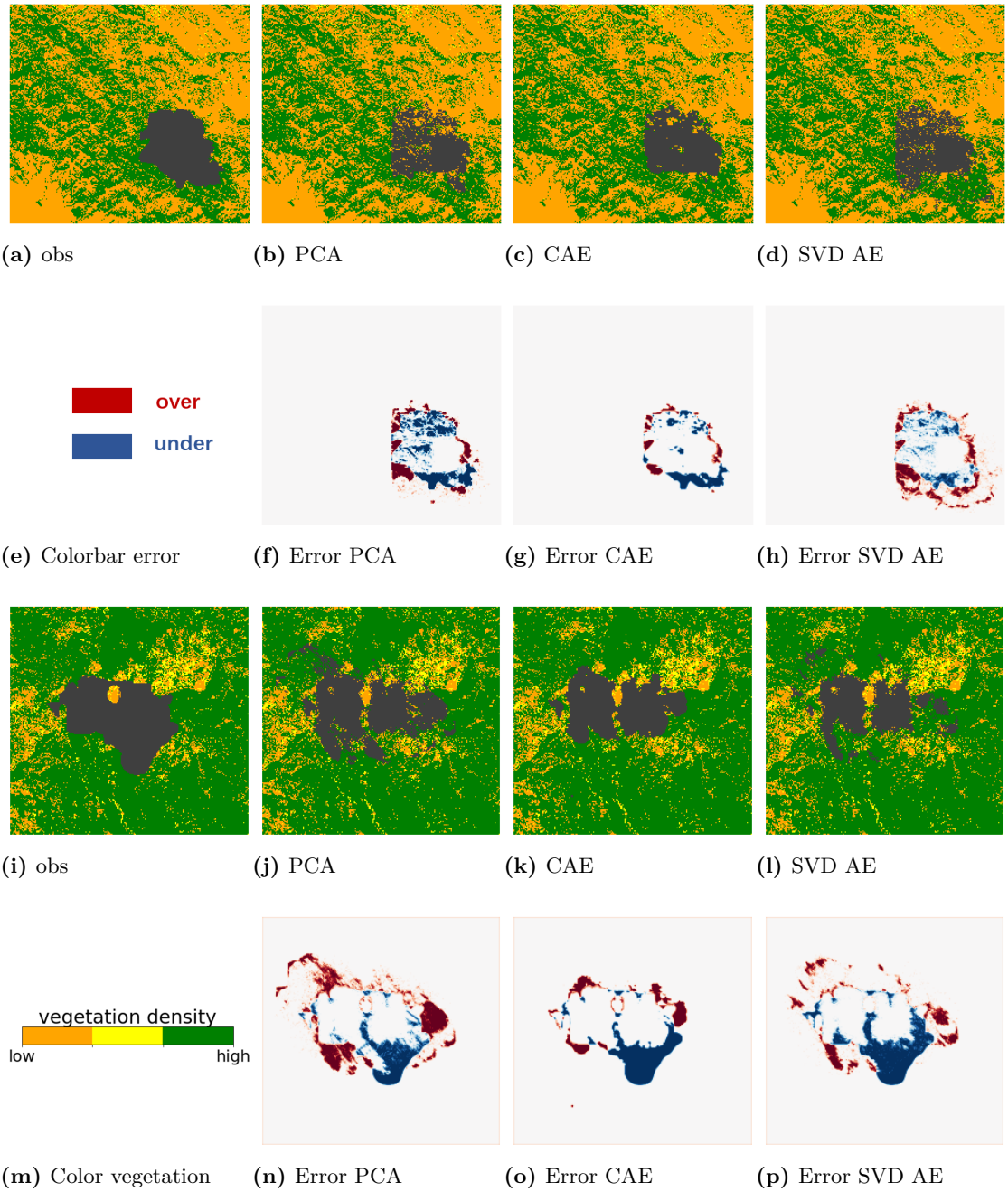
(e) Colorbar error **(f)** Error PCA **(g)** Error CAE **(h)** Error SVD AE

(i) CA **(j)** PCA **(k)** CAE **(l)** SVD AE

(m) Color vegetation **(n)** Error PCA **(o)** Error CAE **(p)** Error SVD AE

Figure 6. Reconstruction of CA simulations for 2 examples in the test dataset of the Ferguson fire at day 4

ROMs are trained using simulated data, their adaptation to satellite images is imperfect. Nevertheless, the GLA produces a better forecast of observed burned area (Figure 10 (c,h)) for both CAE-MLP and PCA-MLP approaches. The predictions performed using assimilated parameters (Figure 10 (b,e,g,j)) are much closer to the satellite observations (Figure 10 (c,h)) regarding initial guesses (Figure 10 (a,d,f,i)). Quantitative comparison of averaged relative error (equation (24)) is shown in Table 6. A considerable reduction of prediction error for all combinations of ML and ROM approaches can be observed. This is coherent with our analysis of Figures 10 and 7. Thus, GLA preforms well in correcting both the overestimation (Ferguson fire) and underestimation (Chimney fire) of model parameters. The predictions of future burned area of day 6 made by assimilated parameters are much closer to the observations than the predictions using prior parameters. Further more, the



(a) obs **(b)** PCA **(c)** CAE **(d)** SVD AE

(e) Colorbar error **(f)** Error PCA **(g)** Error CAE **(h)** Error SVD AE

(i) obs **(j)** PCA **(k)** CAE **(l)** SVD AE

(m) Color vegetation **(n)** Error PCA **(o)** Error CAE **(p)** Error SVD AE

Figure 7. Reconstruction of the satellite observation for the Chimney (a-h) and the Ferguson (i-p) fire at day 4

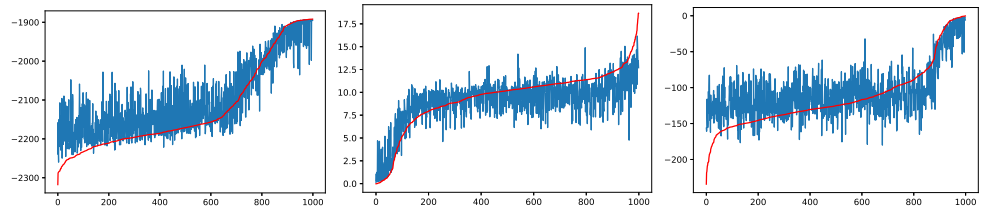
low computational time of GLA (evaluated on the test dataset since only one satellite observation is available for each fire at day 6), allows efficient online parameter estimation, facilitating near real-time fire assimilation/monitoring.

Table 5. Averaged computational time for offline training (on the training dataset of 1000 samples) and online prediction(for each sample including decoding) for different ROMs using RF. The dimension of the latent space is fixed as $q = 30$.

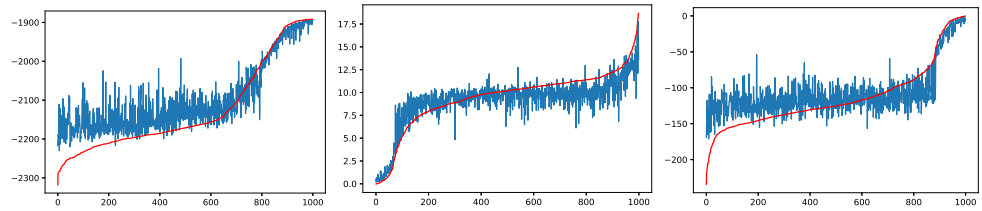
Fire	offline ROM and ML training						online prediction			
	PCA	CAE	SVD AE	KNN	RF	MLP	CA	PCA	CAE	SVD AE
Chimney	101.06s	≈ 1h38min	414.55s	0.02s	0.67s	108s	≈ 35min	0.52s	0.41s	14.47s
Ferguson	97.50s	≈ 1h29min	316.61s	0.02s	0.54s	116s	≈ 29min	0.25S	0.31s	18.62s

Table 6. Relative prediction error and the averaged computational time (only the LA for parameter estimation) for the Chimney and the Ferguson fire at day 6 with assimilated parameters.

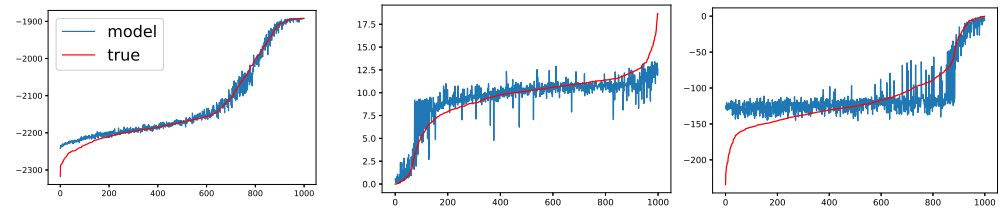
Fire	Data	Forward	PCA			CAE		
			prior	posterior	time	prior	posterior	time
Chimney	CA(test)	KNN	10.1%	6.0%	0.98s	11.1%	7.5%	0.52s
		RF	10.4%	5.7%	0.78s	10.2%	6.6%	0.45s
		MLP	10.9%	6.3%	1.25s	10.6%	5.0%	0.46s
	observation	MLP	8.3%	5.0%		9.65%	6.5%	
Ferguson	CA(test)	KNN	30.5%	20.3%	0.74s	21.3%	9.4%	0.60s
		RF	27.0%	22.5%	0.60s	23.1%	13.6%	0.58s
		MLP	30.7%	17.9%	0.76s	22.4%	12.6%	0.88s
	observation	MLP	41.6%	14.8%		23.8%	11.9%	



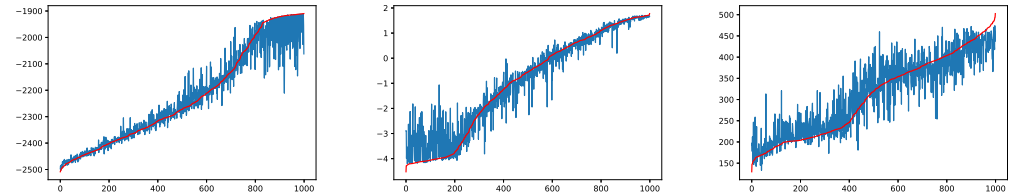
(a) PCA-KNN (C) **(b)** CAE-KNN (C) **(c)** SVD AE-KNN (C)



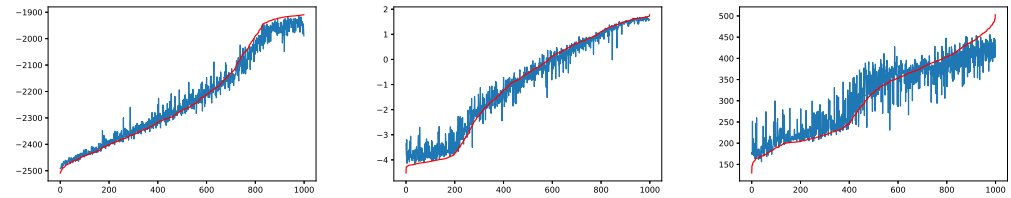
(d) PCA-RF (C) **(e)** CAE-RF (C) **(f)** SVD AE-RF (C)



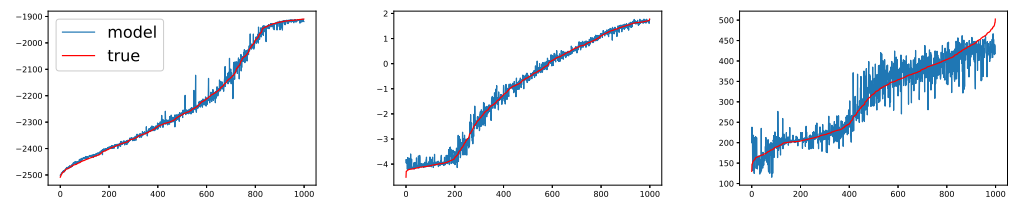
(g) PCA-MLP (C) **(h)** CAE-MLP (C) **(i)** SVD AE-MLP (C)



(j) PCA-KNN (F) **(k)** CAE-KNN (F) **(l)** SVD AE-KNN (F)

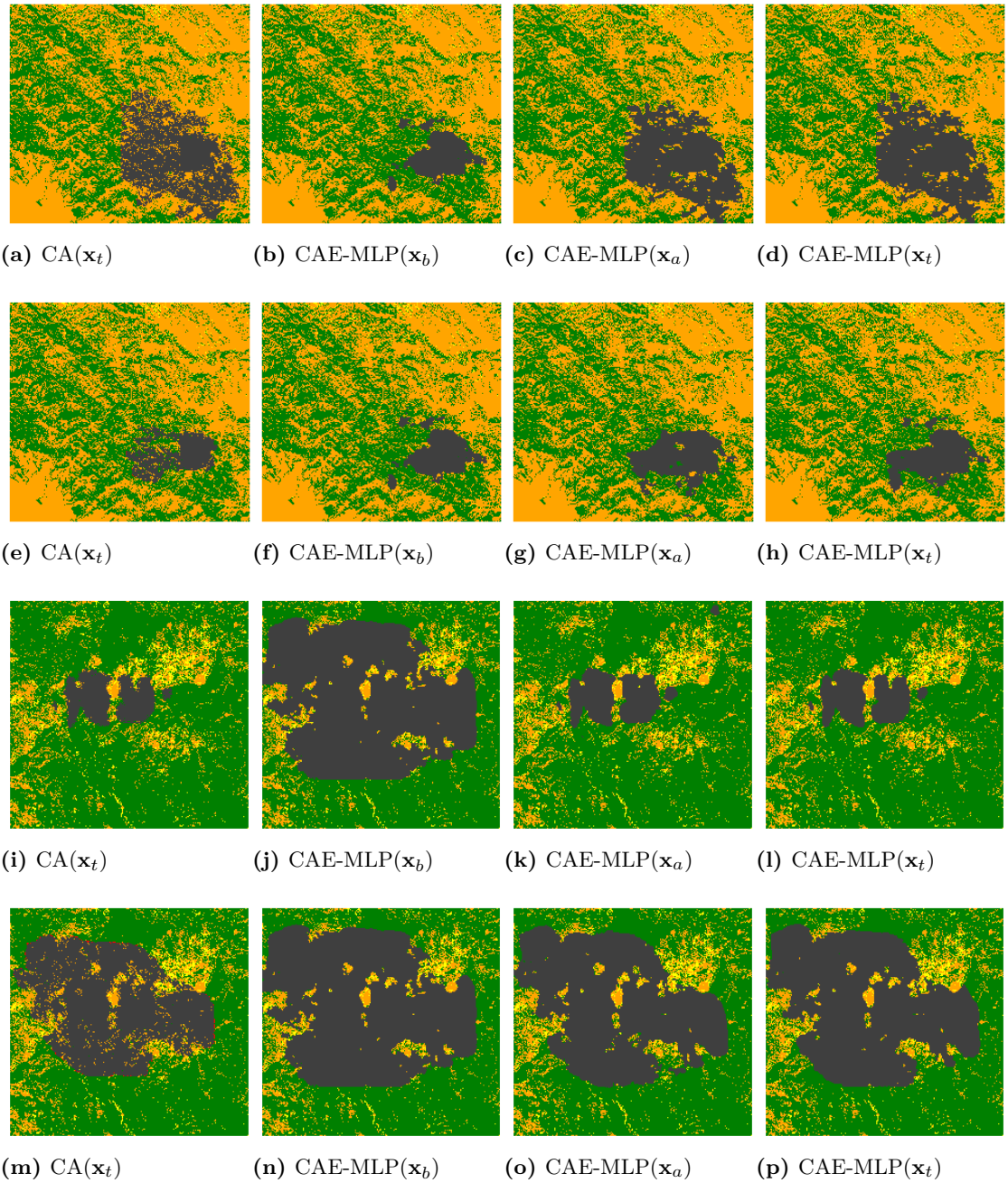


(m) PCA-RF (F) **(n)** CAE-RF (F) **(o)** SVD AE-RF (F)



(p) PCA-MLP (F) **(q)** CAE-MLP (F) **(r)** SVD AE-MLP (F)

Figure 8. Prediction results of latent variables (with the highest variance in each case) for $q = 30$ in the test dataset for both Chimney(C) and Ferguson(F) wildfires at day 4. The x-axis represents reordered samples according to their true latent values



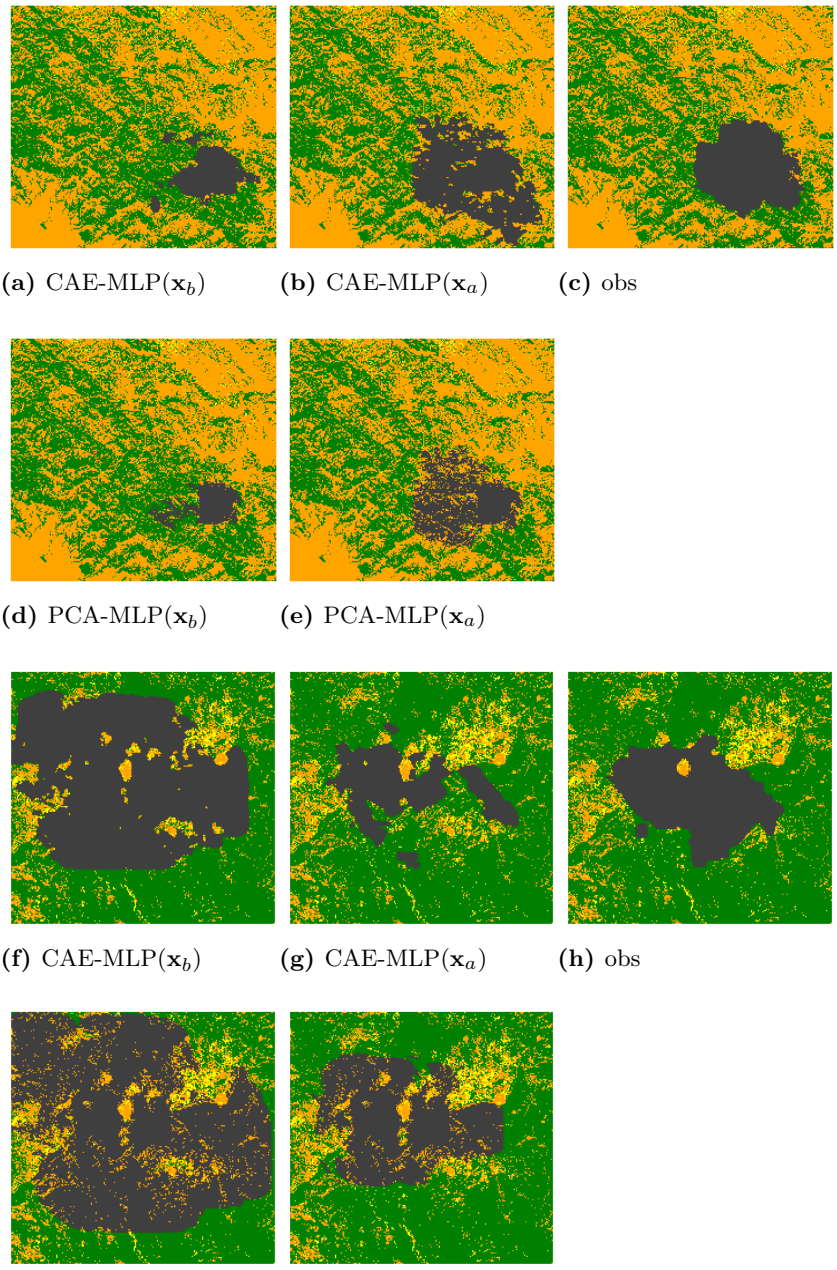
(a) CA($\mathbf{x}_t$) **(b)** CAE-MLP($\mathbf{x}_b$) **(c)** CAE-MLP($\mathbf{x}_a$) **(d)** CAE-MLP($\mathbf{x}_t$)

(e) CA($\mathbf{x}_t$) **(f)** CAE-MLP($\mathbf{x}_b$) **(g)** CAE-MLP($\mathbf{x}_a$) **(h)** CAE-MLP($\mathbf{x}_t$)

(i) CA($\mathbf{x}_t$) **(j)** CAE-MLP($\mathbf{x}_b$) **(k)** CAE-MLP($\mathbf{x}_a$) **(l)** CAE-MLP($\mathbf{x}_t$)

(m) CA($\mathbf{x}_t$) **(n)** CAE-MLP($\mathbf{x}_b$) **(o)** CAE-MLP($\mathbf{x}_a$) **(p)** CAE-MLP($\mathbf{x}_t$)

Figure 9. Machine learning (DL) prediction of burned area at day 6 for both fire events (Chimney a-h and Ferguson i-p) using original ($\mathbf{x}_b$) and assimilated ($\mathbf{x}_a$) parameters compared to CA simulations (considered as ground truth here) for 2 examples in the test dataset.



(a) CAE-MLP($\mathbf{x}_b$) **(b)** CAE-MLP($\mathbf{x}_a$) **(c)** obs

(d) PCA-MLP($\mathbf{x}_b$) **(e)** PCA-MLP($\mathbf{x}_a$)

(f) CAE-MLP($\mathbf{x}_b$) **(g)** CAE-MLP($\mathbf{x}_a$) **(h)** obs

(i) PCA-MLP($\mathbf{x}_b$) **(j)** PCA-MLP($\mathbf{x}_a$)

Figure 10. Machine learning (DL) prediction of burned area at day 6 for both fire events using original ($\mathbf{x}_b$) and assimilated ($\mathbf{x}_a$) parameters compared to satellite observations. The results of both CAE-MLP (a,b,f,g) and PCA-MLP (d,e,i,j) are demonstrated.

5. Conclusion and future work

Current tools for dynamical wildfire forecasting are difficult to apply for real-time fire forecasting. We have developed a scheme which combines different reduced-order modellings and data-driven prediction models for efficient parameter-flexible burned area forecasting. We have extended the Latent Assimilation framework to time-variant (i.e., 4Dvar) optimisation problems. Our parameter estimation approach can be adjusted efficiently using available observation data in near real-time. The results clearly demonstrate the efficiency and the robustness of the proposed approach. The method relies on ML from physics-based CA simulations; such simulations have been successfully run for many different regions (e.g., [57,58]) and thus our approach should be generalisable to other regions and ecosystems. The system represented in this study is also data-agnostic, and could easily be applied to other dynamical systems.

Nevertheless, although the reduced-order modelling and the forward prediction methods achieved a precise reconstruction on the test dataset (also generated by the CA model), there are still difficulties in predicting real-time satellite images. Further efforts to improve the adaptive capability of the current system when dealing unseen satellite data could focus on, for instance, enhancing the regularity of autoencoding via variational autoencoders [17], online fine tuning [59] and domain adaption techniques [60]. Our approach requires individual ROMs and ML prediction models for different ecoregions, which could be computationally expensive for offline simulation and training. It may be possible to overcome this limitation, by using fire prediction models on a global scale with sparser grids, which yields a tradeoff between forecasting accuracy and generalisability. Further work to improve the forward and inverse modellings could focus on learning from more complex fire simulations, for example simulations which take into account of fuel moisture, the incidence of spotting or surface-to-crown transitions

Acknowledgements

The authors would like to thank Mr.Yuhan Huang for performing satellite data curation. The authors thank the anonymous reviewers for their careful reading of our manuscript and their many insightful comments and suggestions. 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 UncERtainty for MultiphasE Systems (PREMIERE).

Acronyms

NN	Neural Network
ML	Machine Learning
LA	Latent Assimilation
DA	Data Assimilation
PR	Polynomial Regression
AE	Autoencoder
VAE	Variational Autoencoder
CAE	Convolutional Autoencoder
VAE	Variational Autoencoder
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	long short-term memory
POD	Proper Orthogonal Decomposition
PCA	Principal Component Analysis
PC	principal component
SVD	Singular Value Decomposition
ROM	reduced-order modelling
CFD	computational fluid dynamics
NWP	numerical weather prediction

MSE	mean square error
LHS	Latin Hypercube Sampling
DL	Deep Learning
KNN	K-nearest Neighbours
DT	Decision Tree
RF	Random Forest
KF	Kalman filter
CART	Classification And Regression Tree
CA	Cellular Automata
MLP	Multi Layer Percepton
GLA	Generalised Latent Assimilation
4Dvar	Four-dimensional variational data assimilation
MODIS	Moderate Resolution Imaging Spectroradiometer
VIIRS	Visible Infrared Imaging Radiometer Suite

1. Chen, G.; Guo, Y.; Yue, X.; Tong, S.; Gasparrini, A.; Bell, M.L.; Armstrong, B.; Schwartz, J.; Jaakkola, J.J.; Zanobetti, A.; et al. Mortality risk attributable to wildfire-related PM2 · 5 pollution: a global time series study in 749 locations. *The Lancet Planetary Health* **2021**, *5*, e579–e587.
2. Verisk Wildfire Risk Analysis: Number of properties at high to extreme risk from wildfire **2021**.
3. Hanson, H.P.; Bradley, M.M.; Bossert, J.E.; Linn, R.R.; Younker, L.W. The potential and promise of physics-based wildfire simulation. *Environmental Science & Policy* **2000**, *3*, 161–172.
4. Viegas, D.X.; Simeoni, A. Eruptive behaviour of forest fires. *Fire technology* **2011**, *47*, 303–320.
5. Valero, M.M.; Jofre, L.; Torres, R. Multifidelity prediction in wildfire spread simulation: Modeling, uncertainty quantification and sensitivity analysis. *Environmental Modelling & Software* **2021**, *141*, 105050.
6. Maffei, C.; Menenti, M. Predicting forest fires burned area and rate of spread from pre-fire multispectral satellite measurements. *ISPRS Journal of Photogrammetry and Remote Sensing* **2019**, *158*, 263–278.
7. Alexandridis, A.; Vakalis, D.; Siettos, C.; Bafas, G. A cellular automata model for forest fire spread prediction: The case of the wildfire that swept through Spetses Island in 1990. *Applied Mathematics and Computation* **2008**, *204*, 191–201.
8. Papadopoulos, G.D.; Pavlidou, F.N. A comparative review on wildfire simulators. *IEEE systems Journal* **2011**, *5*, 233–243.
9. Casas, C.Q.; Arcucci, R.; Wu, P.; Pain, C.; Guo, Y.K. A Reduced Order Deep Data Assimilation model. *Physica D: Nonlinear Phenomena* **2020**, *412*, 132615.
10. Murata, T.; Fukami, K.; Fukagata, K. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics* **2020**, *882*.
11. Ravuri, S.; Lenc, K.; Willson, M.; Kangin, D.; Lam, R.; Mirowski, P.; Fitzsimons, M.; Athanassiadou, M.; Kashem, S.; Madge, S.; et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature* **2021**, *597*, 672–677.
12. Gong, H.; Cheng, S.; Chen, Z.; Li, Q. Data-Enabled Physics-Informed Machine Learning for Reduced-Order Modeling Digital Twin: Application to Nuclear Reactor Physics. *Nuclear Science and Engineering* **2022**, pp. 1–26.
13. Fukami, K.; Murata, T.; Zhang, K.; Fukagata, K. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics* **2021**, *926*.
14. Hinze, M.; Volkwein, S. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In *Dimension reduction of large-scale systems*; Springer, 2005; pp. 261–306.
15. Cheng, S.; Lucor, D.; Argaud, J.P. Observation data compression for variational assimilation of dynamical systems. *Journal of Computational Science* **2021**, p. 101405.
16. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48.
17. Pu, Y.; Gan, Z.; Henao, R.; Yuan, X.; Li, C.; Stevens, A.; Carin, L. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems* **2016**, *29*.

18. Phillips, T.R.; Heaney, C.E.; Smith, P.N.; Pain, C.C. An autoencoder-based reduced-order model for eigenvalue problems with application to neutron diffusion. *International Journal for Numerical Methods in Engineering* **2021**, *122*, 3780–3811.
19. Quilodrán-Casas, C.; Arcucci, R.; Mottet, L.; Guo, Y.; Pain, C. Adversarial autoencoders and adversarial LSTM for improved forecasts of urban air pollution simulations. *arXiv preprint arXiv:2104.06297* **2021**.
20. Pache, R.; Rung, T. Data-driven surrogate modeling of aerodynamic forces on the superstructure of container vessels. *Engineering Applications of Computational Fluid Mechanics* **2022**, *16*, 746–763.
21. Ly, H.; Tran, H. Modeling and control of physical processes using proper orthog- onal decomposition. *J Math Comput Model* **2001**, *33*, 223–236.
22. Xiao, D.; Du, J.; Fang, F.; Pain, C.; Li, J. Parameterised non-intrusive reduced order methods for ensemble Kalman filter data assimilation. *Computers & Fluids* **2018**, *177*, 69–77.
23. Xiao, D.; Fang, F.; Pain, C.; Navon, I. A parameterized non-intrusive reduced order model and error analysis for general time-dependent nonlinear partial differential equations and its applications. *Computer Methods in Applied Mechanics and Engineering* **2017**, *317*, 868–889.
24. Audouze, C.; Vuyst, F.D.; Nair, P.B. Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis. *International Journal for Numerical Methods in Engineering* **2009**, *80*.
25. Xiao, D.; Yang, P.; Fang, F.; Xiang, J.; Pain, C.C.; Navon, I.M. Non-intrusive reduced order modelling of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering* **2016**, *303*, 35–54.
26. Liu, C.; Fu, R.; Xiao, D.; Stefanescu, R.; Sharma, P.; Zhu, C.; Sun, S.; Wang, C. EnKF data-driven reduced order assimilation system. *Engineering Analysis with Boundary Elements* **2022**, *139*, 46–55.
27. Cheng, S.; Prentice, I.C.; Huang, Y.; Jin, Y.; Guo, Y.K.; Arcucci, R. Data-driven surrogate model with latent data assimilation: Application to wildfire forecasting. *Journal of Computational Physics* **2022**, p. 111302.
28. Finney, M.A. *FARSITE, Fire Area Simulator–model development and evaluation*; Number 4, US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.
29. Hilton, J.E.; Sullivan, A.L.; Swedosh, W.; Sharples, J.; Thomas, C. Incorporating convective feedback in wildfire simulations using pyrogenic potential. *Environmental Modelling & Software* **2018**, *107*, 12–24.
30. Mallet, V.; Keyes, D.E.; Fendell, F. Modeling wildland fire propagation with level set methods. *Computers & Mathematics with Applications* **2009**, *57*, 1089–1101.
31. Muñoz-Esparza, D.; Kosović, B.; Jiménez, P.A.; Coen, J.L. An accurate fire-spread algorithm in the Weather Research and Forecasting model using the level-set method. *Journal of Advances in Modeling Earth Systems* **2018**, *10*, 908–926.
32. Ambroz, M.; Mikula, K.; Fraštia, M.; Marčiš, M. Parameter estimation for the forest fire propagation model. *Tatra Mountains Mathematical Publications* **2018**, *75*, 1–22.
33. Lautenberger, C. Wildland fire modeling with an Eulerian level set method and automated calibration. *Fire Safety Journal* **2013**, *62*, 289–298.
34. Rochoux, M.; Emery, C.; Ricci, S.; Cuenot, B.; Trouvé, A. A comparative study of parameter estimation and state estimation approaches in data-driven wildfire spread modeling. In Proceedings of the VII International Conference on Forest Fire Research, Coimbra (Portugal), 2014, pp. 14–20.
35. Ervilha, A.; Pereira, J.; Pereira, J. On the parametric uncertainty quantification of the Rothermel's rate of spread model. *Applied Mathematical Modelling* **2017**, *41*, 37–53.
36. Zhang, C.; Collin, A.; Moireau, P.; Trouvé, A.; Rochoux, M.C. State-parameter estimation approach for data-driven wildland fire spread modeling: Application to the 2012 RxCADRE S5 field-scale experiment. *Fire Safety Journal* **2019**, *105*, 286–299.
37. Alessandri, A.; Bagnerini, P.; Gaggero, M.; Mantelli, L. Parameter estimation of fire propagation models using level set methods. *Applied Mathematical Modelling* **2021**, *92*, 731–747.
38. Jensen, C.A.; Reed, R.D.; Marks, R.J.; El-Sharkawi, M.A.; Jung, J.B.; Miyamoto, R.T.; Anderson, G.M.; Eggen, C.J. Inversion of feedforward neural networks: algorithms and applications. *Proceedings of the IEEE* **1999**, *87*, 1536–1549.
39. Amendola, M.; Arcucci, R.; Mottet, L.; Casas, C.Q.; Fan, S.; Pain, C.; Linden, P.; Guo, Y.K. Data Assimilation in the Latent Space of a Neural Network, 2020, [arXiv:cs.LG/2012.12056].

40. Peyron, M.; Fillion, A.; Gürol, S.; Marchais, V.; Gratton, S.; Boudier, P.; Goret, G. Latent space data assimilation by using deep learning. *Quarterly Journal of the Royal Meteorological Society* **2021**, *147*, 3759–3777.
41. Cheng, S.; Chen, J.; Anastasiou, C.; Angeli, P.; Matar, O.K.; Guo, Y.K.; Pain, C.C.; Arcucci, R. Generalised Latent Assimilation in Heterogeneous Reduced Spaces with Machine Learning Surrogate Models. *arXiv preprint arXiv:2204.03497* **2022**.
42. Carrassi, A.; Bocquet, M.; Bertino, L.; Evensen, G. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change* **2018**, *9*, e535.
43. Cheng, S.; Argaud, J.P.; Iooss, B.; Lucor, D.; Ponçot, A. Background error covariance iterative updating with invariant observation measures for data assimilation. *Stochastic Environmental Research and Risk Assessment* **2019**, *33*, 2033–2051.
44. Ide, K.; Courtier, P.; Ghil, M.; Lorenc, A.C. Unified notation for data assimilation: Operational, sequential and variational (gtspecial issueltdata assimilation in meteology and oceanography: Theory and practice). *Journal of the Meteorological Society of Japan. Ser. II* **1997**, *75*, 181–189.
45. Drury, S.A.; Rauscher, H.M.; Banwell, E.M.; Huang, S.; Lavezzo, T.L. The interagency fuels treatment decision support system: functionality for fuels treatment planning. *Fire Ecology* **2016**, *12*, 103–123.
46. Weise, D.R.; Biging, G.S. A qualitative comparison of fire spread models incorporating wind and slope effects. *Forest Science* **1997**, *43*, 170–180.
47. Hersbach, H. Copernicus Climate Change Service (C3S) (2017): ERA5: Fifth generation of ECMWF atmospheric reanalyses of the global climate. Technical report, Copernicus Climate Change Service Climate Data Store (CDS), 2017.
48. Giglio, L.; Schroeder, W.; Justice, C.O. The collection 6 MODIS active fire detection algorithm and fire products. *Remote Sensing of Environment* **2016**, *178*, 31–41.
49. Schroeder, W.; Oliva, P.; Giglio, L.; Csiszar, I.A. The New VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment* **2014**, *143*, 85–96.
50. Scaduto, E.; Chen, B.; Jin, Y. Satellite-Based Fire Progression Mapping: A Comprehensive Assessment for Large Fires in Northern California. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2020**, *13*, 5102–5114.
51. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation* **2017**, *29*, 2352–2449.
52. Lewis, R.J. An introduction to classification and regression tree (CART) analysis. In Proceedings of the Annual meeting of the society for academic emergency medicine in San Francisco, California. Citeseer, 2000, Vol. 14.
53. Ram, P.; Sinha, K. Revisiting kd-tree for nearest neighbor search. In Proceedings of the Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining, 2019, pp. 1378–1388.
54. Tandeo, P.; Ailliot, P.; Bocquet, M.; Carrassi, A.; Miyoshi, T.; Pulido, M.; Zhen, Y. A Review of Innovation-Based Methods to Jointly Estimate Model and Observation Error Covariance Matrices in Ensemble Data Assimilation. *arXiv preprint arXiv:1807.11221, accepted for submission to Monthly Weather Review* **2018**.
55. Cheng, S.; Qiu, M. Observation error covariance specification in dynamical systems for data assimilation using recurrent neural networks. *Neural Computing and Applications* **2021**, pp. 1–19.
56. Fulton, W. Eigenvalues, invariant factors, highest weights, and Schubert calculus. *Bulletin of The American Mathematical Society* **2000**, *37*, 209–250.
57. Trucchia, A.; D'Andrea, M.; Baghino, F.; Fiorucci, P.; Ferraris, L.; Negro, D.; Gollini, A.; Severino, M. PROPAGATOR: an operational cellular-automata based wildfire simulator. *Fire* **2020**, *3*, 26.
58. Freire, J.G.; DaCamara, C.C. Using cellular automata to simulate wildfire propagation and to assist in fire management. *Natural hazards and earth system sciences* **2019**, *19*, 169–179.
59. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* **2018**.
60. Peng, X.; Huang, Z.; Zhu, Y.; Saenko, K. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054* **2019**.