Data-Driven Surrogate Model with Latent Data Assimilation: Application to Wildfire Forecasting

Sibo Cheng, I Colin Prentice, Yuhan Huang, Yufang Jin, Yi-Ke Guo, Rossella Arcucci

Data-Driven Surrogate Model with Latent Data Assimilation: Application to Wildfire Forecasting

Sibo Cheng[1,2], I. Colin Prentice[2], Yuhan Huang[3], Yufang Jin[3], Yi-Ke Guo[1], Rossella Arcucci[4,1]

[1] *Data Science Instituite, 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] *Department of Earth Science & Engineering, Imperial College London, UK*

Abstract

The large and catastrophic wildfires have been increasing across the globe in the recent decade, highlighting the importance of simulating and forecasting fire dynamics in near real-time. This is extremely challenging due to the complexities of physical models and geographical features. Running physics-based simulations for large wildfire events in near real-time is computationally expensive, if not infeasible. In this work, we develop and test a novel data-model integration scheme for fire progression forecasting, that combines Reduced-order modelling, recurrent neural networks (Long-Short-Term Memory), data assimilation, and error covariance tuning. The Reduced-order modelling and the machine learning surrogate model ensure the efficiency of the proposed approach while the data assimilation enables the system to adjust the simulation with observations. We applied this algorithm to simulate and forecast three recent large wildfire events in California from 2017 to 2020. The deep-learning-based surrogate model runs around 1000 times faster than the Cellular Automata simulation which is used to generate training data-sets. The daily fire perimeters derived from satellite observation are used as observation data in Latent Assimilation to adjust the fire forecasting in near real-time. An error covariance tuning algorithm is also performed in the reduced space to estimate prior simulation and observation errors. The evolution of the averaged relative root mean square error (R-RMSE) shows that data assimilation and covariance tuning reduce the RMSE by about 50% and considerably improves the forecasting accuracy. As a first attempt at a fast reduced order wildfire spread forecasting, our results highlight the potential of data-driven machine learning models for providing guidance on fire suppression and evacuation efforts.

Keywords: Deep learning, Reduced-order modelling, Data assimilation, Wildfire forecasting, LSTM, Fire spread

1. Introduction

Wildland fires have been recognized as one of the most damaging natural disasters affecting human, plants and animals. These wildfires, often lightning-caused,

Email address: r.arcucci@imperial.ac.uk

have substantially affected the Earth's surface and environment for over 300 million years [29]. According to [1], the number of acres burned annually has increased by three times globally over the past 20 years. The recent wildfires in California in 2018 cost more than $27 billion capital loss while associated air pollution of the corresponding areas affects roughly 31% population of United States [76]. Operating strategies related to firefighting resources allocation, or evacuation of at-risk areas can benefit from numerical models that predict the fire spread in space and time. However, forecasting wildfire dynamics is highly challenging due to the complexities linked to the physical models and the geographical features. Nevertheless, there has been increased attention given to the subject world-widely [55, 66]. The factors behind fire spread behaviour depend heavily on local physical, environmental and meteorological variables [78, 71, 50], such as vegetation density distribution, landscape slope, fuel continuity and wind dynamics. For reasons including high dimensionality and the lack of historical data, among others, current fire modelling systems, for instance, those based on Rothermel equation [67], NWP(numerical weather prediction)-driven forecasting and Cellular Automation (CA)[2], remain highly empirical [58]. System parameters are often obtained using calibrations of local fire scenarios [14, 15]. Furthermore, running physics-based simulations for large-scale wildfires can be computationally expensive, even when coupled with discrete event modelling (e.g., Discrete EVent System [44], Cellular Automata [56]) and parallelization computing, leading to difficulties for real-time fire progression monitoring [64]. Much effort has been devoted to improving the prediction accuracy and efficiency via data-driven surrogate models, including the use of machine learning (ML) [49] and data assimilation (DA) techniques [52]. The latter, initially developed for NWP, manages to combine the information embedded in the model simulations and the time series observation data [8], for instance, from satellite images, with an optimal weight regarding the quantification of prior uncertainties.

In order to manage large datasets of wildfire modellings, which are often multivariate and multidimensional, reduced-order modelling (ROM) techniques have been widely adopted in quantitative fire studies. For example, the proper orthogonal decomposition technique (POD) has been used to describe fire dynamics with a significantly reduced state space, regarding the original problem [39, 40]. Recently, convolutional autoencoders have also been employed to identify regions with a high likelihood of wildfires [46].

In this study, we aim to enable real-time integration of the observation data for improving predictions provided by the data-driven surrogate model by combining machine learning, model reduction and data assimilation techniques to enhance the efficiency and accuracy of the fire monitoring system. Our objective is to promptly predict the evolution of burned areas to indicate the development of wildfires. Therefore, performing ML prediction and DA at a low cost is of prime concern. As shown in Figure 1, we first train a Long-Short-Term-Memory (LSTM) neural network (a type of RNN, efficient to handle long-term dependencies [13]) in the low-dimensional latent space over a CA-based stochastic simulator [2] incorporating actual landscape data, including canopy density/cover and landscape slope. Real-time satellite images are then used to adjust the current prediction through a DA scheme. Since both the machine learning prediction and the DA correction are performed in the

latent space, advanced covariance tuning approaches (e.g., [28, 26]) which are usually resource/memory intensive in the full space [75] can be applied to enhance the DA precision with a reduced computational cost.

The specification of these error covariance matrices, which update the optimal weighting between model simulations and instrumental observations through prior error covariances iteratively, is crucial for the reconstruction/prediction of dynamical systems, such as wildfire progression. By integrating DA algorithms for real-time estimation based on time series observation data, the prediction accuracy of the data-driven model can be significantly improved compared to the pure ML approach.

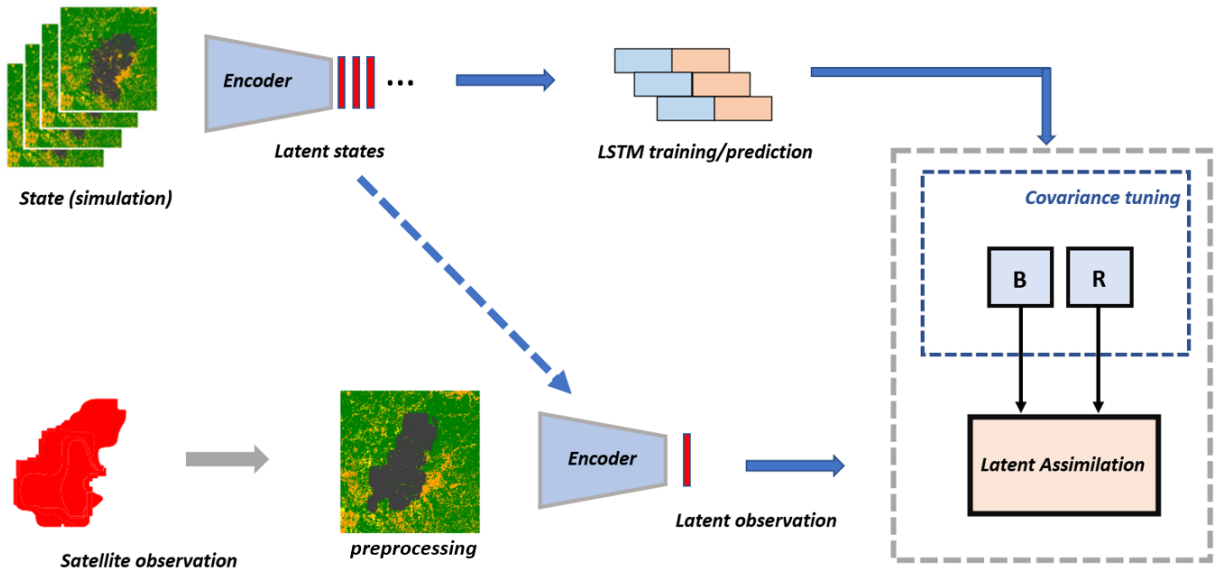


Figure 1: Flowchart of deep latent assimilation with information-based observation compression and error covariance tuning

2. Related work and contribution

Given the sharp increase in the amount of available environmental data, the application of ML approaches, including random forest, reinforcement learning, deep neural networks, has been implemented in a variety of wildfire-related problems, for instance, fire detection[83, 10], fire occurrence prediction[30], and fire susceptibility mapping [82]. As an example, both convolutional neural networks (CNN) and support vector machines (SVM) have been adapted for wildfire detection [83, 10] relying on satellite images of fire and smoke. [9] employs genetic algorithms (GA) for optimal sensor placement in wildfire monitoring. [47] builds a hybrid model which combines GA and neuro-fuzzy systems to predict wildfire probability in a spatially explicit way. For other applications of ML methods in wildfire-related studies, interested readers are referred to the review of [49]. Here, we focus on predicting fire growth and burned surface, which is a crucial task for fire safety science. [53] implements a range of ML classifiers based on satellite image recognition to predict if a fire will grow rapidly. They quote a success rate to identify large fires of over 75%. Similar methodology can be found in [51, 3] for predicting the final burned area in a wildfire. Likewise, [35] adopts the reinforcement learning method, which

uses satellite images to define the reward function, for predicting the fire dynamics through agent-based policies. The recent work of [84] makes use of a deep neural network (DNN) to surrogate the energy exascale Earth system model for predicting the area of burned surface. However, to the best of author's knowledge, due to the high dimensionality and a large number of local environmental variables, there have been no purely data-driven regression approaches developed which consider land characteristics to forecast the fire front/burned surface dynamics on a continuous basis. On the other hand, a number of physics-based fire simulation tools, such as FARSITE[32], SPARK[42], have been widely used to simulate the diffusion of wildfires incorporating geophysical features[56]. However, these simulation tools require several layers of geophysical information, including canopy cover/density, fuel, slope and wind speed, which are costly to build for real-world problems[56]. Building surrogate models in the latent space regarding the spatial variance of physics-informed simulations can be a viable solution to decrease the computational burden.

Real-time observations, such as satellite images, are often used to adjust prior simulation/prediction of wildfire spread. A variety of inverse modelling approaches [48, 65, 45, 63], including data assimilation techniques, have been adopted for updating fire progression estimation and forecasting. The work of [48] employs a tangent linear model on top of DA algorithms to enhance the simulation accuracy of building fire modelling. [65] proposes the use of ensemble assimilation methods to adjust the fire front forecasted via a polynomial chaos (PC) expansion in synthetically generated wildfire scenarios. Usually, these DA corrections take place either in the full-physical space or their sub-domains where real-time adjusting remains cumbersome due to the large dimension of the physical fields.

A key challenge of wildfire forecasting is to develop (surrogate) models, capable of real-time fire forecasting/nowcasting/adjusting which takes advantage of the increasing volume of observation data. Therefore, learning from existing wildfire simulations in an optimal compressed state space is a promising solution that ensures both the accuracy, and more importantly, the efficiency of the surrogate model. Similar ideas of applying machine learning algorithms, namely recurrent neural networks in the reduced latent space have been implemented in a wide range of applications such as computational fluid mechanics (CFD)[68, 7], hydrology[19] and air pollution quantification[16]. The combination of DA, ROM, and ML, more specifically deep learning, has aroused increasing research interest in recent years. The work of [16, 7] posits learning assimilated results via an LSTM network which implies a significant reduction of forecasting errors, evaluated in untrained datasets. The work of [16] was recently extended in [61] which makes use of an Adversarial-Trained LSTM when insufficient data are available for training. The recurrent Kalman network in the latent space is proposed in [11] to make locally linear predictions with factorized covariance matrices to reduce computational cost. The recent work of [4] introduces the concept of Latent Assimilation where a convolutional autoencoder (CAE) network is used to compress the state variables while the DA updatings also arise in the reduced latent space. In data assimilation, prior errors, often supposed to be Gaussian distributed, can be fully characterized by the first (mean) and the second (covariance) statistical moment. The latter, which decides the "optimal information weight" in the loss function, is a crucial element in DA algorithms [33, 23].

In this study, we make the following contributions:

- We propose a novel algorithm scheme, which combines ROM, RNN (LSTM), DA and error covariance tuning for real-time wildfire forecasting/nowcasting (coupled with satellite observations). The convergence of the error covariance tuning approach, namely the Desroziers & Ivanov [28] tuning algorithm (DI01), has been further investigated in this paper. We provide a mathematical explanation why DI01 diverges when the background matrix and the observation matrix have the same correlation structure in the observation space.

- The ROM (both POD and CAE) and the LSTM are trained on the basis of multiple CA simulations. To the best of our knowledge, we are the first to train ML surrogate models for dynamical fire spread problems with a stochastic simulation code (such as CA). The stochastic diffusion makes the NN convergence more challenging, but it contributes to improving the flexibility of the surrogate model by considering a great number of fire scenarios. The latter is crucial when performing real-time corrections (e.g., via DA) on the surrogate model.

- The proposed data-driven approach also contributes to tackling one of the major bottlenecks of real-time fire forecasting/nowcasting: it is time-consuming and computationally expensive to implement either physics-informed simulations or DA in the full physical fields. According to the experiments of three recent large wildfire events in California (Buck 2017, Pier 2017, Bear 2020) made in this paper, the evolution of the proposed surrogate model is thousands of times faster than either CA or CFD-based (e.g., Flammap) simulations. More importantly, the novel ML-based approach makes the real-time DA easily affordable on a laptop CPU.

- The algorithm scheme proposed in this work can be easily applied/extended to other dynamical systems with either variational or Kalman-type DA. The repository can be found: https://github.com/DL-WG/ML_surrogate_model_wildfires

The paper is organized as follows. In Section 3, the principle of DA is briefly introduced with a specific attention on error covariance tuning. We then address the computation of the latent space via either POD or CAE in Section 4, followed by the introduction of RNN (LSTM) surrogate model in Section 5. The application of these methods to wildfire modelling is described in Section 7 and the results are illustrated in Section 8. We close the paper with a discussion.

3. Data assimilation and error covariance tuning

3.1. Data assimilation

Data assimilation algorithms aim to improve the estimation of state variables $\mathbf{x}$ (which can be some physical fields or a set of parameters) based on a prior simulation/forecast $\mathbf{x}_b$ and real-time observations, embedded in the observation vector $\mathbf{y}$. The exact value of the current state, often out of reach in real-world applications, is

denoted by $\mathbf{x}_{\text{true}}$, known as the true state. Prior background and observation errors, respectively denoted as ϵ_b and ϵ_y are supposed to be centred Gaussian with

$$\epsilon_b = \mathbf{x}_b - \mathbf{x}_{\text{true}}, \quad \epsilon_y = \mathcal{H}(\mathbf{x}_{\text{true}}) - \mathbf{y}. \tag{1}$$

where $\mathbf{B}$ and $\mathbf{R}$ are the associated error covariance matrices, defined as

$$\mathbf{B} = \text{Cov}(\epsilon_b, \epsilon_b), \quad \mathbf{R} = \text{Cov}(\epsilon_y, \epsilon_y). \tag{2}$$

In this study, both $\mathbf{x}_b$ and $\mathbf{y}$, representing respectively the model prediction and the satellite observations, are in the compressed/encoded latent space. DA algorithms aim to find a least square estimate of $\mathbf{x}_{\text{true}}$ through the cost function J defined as

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x})) \tag{3}$$

$$= \frac{1}{2}||\mathbf{x} - \mathbf{x}_b||^2_{\mathbf{B}^{-1}} + \frac{1}{2}||\mathbf{y} - \mathcal{H}(\mathbf{x})||^2_{\mathbf{R}^{-1}}, \tag{4}$$

thanks to the observation operator $\mathcal{H}$. The inverse of these covariance matrices $\mathbf{B}^{-1}$ and $\mathbf{R}^{-1}$ determine the weights of $\mathbf{x}_b$ and $\mathbf{y}$ in the objective function.

Equation (3), also known as the three-dimensional variational (3D-Var) formulation, is a general representation of DA problems when the model error is not considered. The minimization point of equation (3), known as the analysis state, is denoted as $\mathbf{x}_a$, i.e.

$$\mathbf{x}_a = \underset{\mathbf{x}}{\text{argmin}}\Big(J(\mathbf{x})\Big). \tag{5}$$

If the observation operator $\mathcal{H}$ can be approximated by some linear operator $\mathbf{H}$, the minimization problem represented by equation (5) leads to the Best Linearized Unbiased Estimator (BLUE) formulation,

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_b) \tag{6}$$

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B} \tag{7}$$

where $\mathbf{A} = \text{Cov}(\mathbf{x}_a - \mathbf{x}_{\text{true}})$ is the analyzed error covariance and $\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}$ is the Kalman gain matrix. From now on, we denote $\mathbf{H}$ as the linearized observation operator. In practice, the observation space in DA is often a subspace of the state space (e.g. [31],[4]), resulting in a block-diagonal structure of the $\mathbf{H}$ matrix. When $\mathcal{H}$ is highly non-linear the minimization of equation (3) often involves gradient descent algorithms (such as "L-BFGS-B" [34] or adjoint-based [24] numerical techniques).

DA algorithms can be applied to dynamical systems thanks to the transition operator $\mathcal{M}_{t^k \to t^{k+1}}$ which links the state variables of time t^k and t^{k+1},

$$\mathbf{x}_{t^{k+1}} = \mathcal{M}_{t^k \to t^{k+1}}(\mathbf{x}_{t^k}). \tag{8}$$

The forecasting thus depends on the knowledge of transition operator $\mathcal{M}_{t^k \to t^{k+1}}$ and the corrected state at the current time $\mathbf{x}_{a,t^k}$. Typically, the current background state is often given by the forecasting from the previous step, i.e.

$$\mathbf{x}_{b,t^k} = \mathcal{M}_{t^{k-1} \to t^k}(\mathbf{x}_{a,t^{k-1}}). \tag{9}$$

Obviously, a more accurate reanalysis $\mathbf{x}_{a,t^{k-1}}$ leads to a more reliable forecasting $\mathbf{x}_{b,t^k}$. In this work, the forward operator $\mathbf{M}$ is determined by a data-driven surrogate model.

3.2. Error covariance tuning

Error covariance matrices modelling, which determines the importance given to the different information sources in the loss function of equation (4), is an essential element in DA algorithms with regards to the forecasting accuracy. The estimation of these covariances often relies on empirical assumptions, including the use of isotropic correlation kernels [36] and experimentally set error amplitudes (i.e., $\mathrm{Tr}(\mathbf{B})$ and $\mathrm{Tr}(\mathbf{R})$). Continuous efforts have been devoted to improving the error covariance specification in dynamical DA models [28, 27, 57, 20]. The data assimilation in this study takes place in the reduced unit-less latent space where the state dimension is considerably reduced in contrast to the full physical space. As a consequence, neither multi-variate nor multi-dimensional covariance computation is required here. Thus, we mainly focus on the ratio of $\mathrm{Tr}(\mathbf{B})$ and $\mathrm{Tr}(\mathbf{R})$, which is of greatest importance since it directly determines the impact of background and observation data. The Desroziers & Ivanov [28] tuning algorithm (DI01), first introduced in the field of meteorological science, adjusts the observation-error weighting parameters by applying an iterative fixed-point procedure without modifying error correlation structures.

As demonstrated in the work of [74] and [28], the following equalities should be satisfied in a linearized 3D-VAR assimilation system with well specified $\mathbf{B}$ and $\mathbf{R}$ matrices,

$$\begin{aligned}\mathbb{E}\left[J_b(\mathbf{x}_a)\right] &= \frac{1}{2}\mathbb{E}\left[(\mathbf{x}_a-\mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}_a-\mathbf{x}_b)\right] \\ &= \frac{1}{2}\mathrm{Tr}(\mathbf{KH}),\end{aligned} \tag{10}$$

$$\mathbb{E}\left[J_o(\mathbf{x}_a)\right] = \frac{1}{2}\mathbb{E}\left[(\mathbf{y}-\mathbf{H}\mathbf{x}_b)^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x}_b)\right] \tag{11}$$

$$= \frac{1}{2}\mathrm{Tr}(\mathbf{I}-\mathbf{HK}). \tag{12}$$

Based on the diagnostic of equation (12), two indicators of error covariance specification can be defined, that is,

$$s_{b,q} = \frac{2J_b(\mathbf{x}_a)}{\mathrm{Tr}(\mathbf{K}_q\mathbf{H})}, \quad s_{o,q} = \frac{2J_o(\mathbf{x}_a)}{\mathrm{Tr}(\mathbf{I}-\mathbf{H}\mathbf{K}_q)}, \tag{13}$$

where q is the current iteration. Following a Maximum Likelihood analysis [18], the two indicators $s_{b,q}$ and $s_{o,q}$, acting as scaling coefficients, can be employed to establish an iterative tuning algorithm, i.e.,

$$\mathbf{B}_{q+1} = s_{b,q}\mathbf{B}_q, \quad \mathbf{R}_{q+1} = s_{o,q}\mathbf{R}_q. \tag{14}$$

It is worth mentioning that the analysis state $\mathbf{x}_{a,q}$ and the gain matrix $\mathbf{K}_q$ depend on $\mathbf{B}_q$, $\mathbf{R}_q$ and thus on the iterative coefficients $s_{b,q}$, $s_{o,q}$. In practice, a stopping criteria of DI01 could be designed by choosing a minimum threshold of $\mathbf{max}(||s_{b,q}-1||, ||s_{o,q}-1||)$. However, according to [17, 21], the convergence of s_b and s_o can be very fast, especially in the ideal case where the correlation patterns of B and R are significantly different. Thus large iteration number is not required as the first iteration could already provide a reasonably good estimation of $\mathrm{Tr}(\mathbf{B})$ and $\mathrm{Tr}(\mathbf{R})$.

4. Reduced-order modelling

4.1. Proper orthogonal decomposition

Being widely adopted in CFD simulations and industrial applications, the proper orthogonal decomposition (POD) [72] is a classical method of model order reduction using snapshots of dynamical systems, widely applied in engineering problems [6, 22, 38]. Here we remind the principle and some of the most important properties of POD for reduced-order modelling. For some dynamical physical field $\mathbf{u}_t$ of dimension n (i.e., $\dim(\mathbf{u}_t) = n, \forall t$), we consider an ensemble of N snapshots $\{\mathbf{u}_k\}_{k=1}^N \subset \mathbb{R}^n$ which are numerical solutions generated by a wildfire simulation code.

The empirical covariance matrix $\mathbf{C}$ of $\{\mathbf{u}_k\}_{k=1}^N$ (after being flattened to 1D vectors) can be estimated by computing the vector product of each pair of snapshots, i.e.,

$$\mathbf{C}_{i,j} = \frac{1}{N}\langle \mathbf{u}_i, \mathbf{u}_j \rangle, \quad \forall 1 \leq i, j \leq N, \tag{15}$$

where $\langle ., . \rangle$ stands for the vector product of two vectors. We can then compute the POD latent space through the eigenvalues $\{\lambda_i\}_{i=1}^n$ and the corresponding eigenvectors $\{v^i\}_{i=1}^n$ of the $\mathbf{C}$ matrix. Here, $\{\lambda_i\}_{i=1}^n$ are listed in a decreasing order (i.e., $\lambda_{i+1} < \lambda_i, \quad \forall i <\in \{1, ..., n-1\}$). More precisely, the unit vector η_j which represents the j-th POD basis, is given by

$$\eta_j = \sum_{i=1}^{N} \lambda_i v_i^j, \tag{16}$$

where v_i^j denotes the i-th element of the j-th eigenvector. The value of the j-th eigenvalue λ_j describes the relative importance of the j-th POD basis vector. The ensemble of the snapshots is represented by $\boldsymbol{U} \in \mathbb{R}^{n \times N}$ which contains $\mathbf{u}_k$ as its columns, i.e.,

$$\boldsymbol{U}[:, k] = \mathbf{u}_k, \quad k \in [1, .., N]. \tag{17}$$

We can construct an optimal projection space of dimension γ with minimum loss of information by simply keeping the γ first POD basis, that is,

$$\arg \min_{\boldsymbol{W} \in \mathbb{R}^{n \times \gamma}} \|\boldsymbol{U} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{U}\|_F^2 = \boldsymbol{Q}_\gamma = [\eta_1, .., \eta_\gamma] \tag{18}$$

$$\min_{\boldsymbol{W} \in \mathbb{R}^{n \times \gamma}} \|\boldsymbol{U} - \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{U}\|_F^2 = \|\boldsymbol{U} - \boldsymbol{Q}_\gamma\boldsymbol{Q}_\gamma^T\boldsymbol{U}\|_F^2 = \sum_{k=\gamma+1}^{N} \lambda_k \ , \tag{19}$$

where $||.||_F$ is the Frobenius norm and γ is so-called the truncation parameter. The columns of $\boldsymbol{Q}_\gamma$ are also known as vectors of principale components. Among all orthonormal bases of dimension γ (usually $\gamma \ll n$), $\mathbf{Q}_\gamma$ minimizes the least squares error of the reconstruction of the snapshot matrix $\boldsymbol{U}$. $\boldsymbol{Q}_\gamma\boldsymbol{Q}_\gamma^T\boldsymbol{U}$ is also known as the reconstruction of the optimal reduced space.

Thus the POD basis yields an orthonormal basis that provides an efficient low-dimensional representation of the dynamical system. The data compression accuracy C_a and compression ratio C_r are defined as:

$$C_a = \sum_{k=\gamma+1}^{N} \lambda_k \Big/ \sum_{k=1}^{N} \lambda_k, \quad C_r = \gamma/N, \tag{20}$$

both ranging from 0 to 1 by their definitions. For any snapshots $\mathbf{u}_t \in \mathbb{R}^{n\times 1}$ converted to a 1D vector, the γ-dimensional representation where the surrogate model predictions take place in this paper, is approximated by

$$\mathbf{x} = \mathbf{u}_t^{\gamma} = \mathbf{Q}_{\gamma}^{T}\mathbf{u}_t \ . \tag{21}$$

4.2. Deep AutoEncoder

Similar to POD, AutoEncoding [12] is a data compression algorithm based on a feed-forward neural network, aiming for reconstructing the input vectors through unsupervised pretraining. More precisely, an AutoEncoder (AE) first encodes the input vector $\mathbf{u}$ as

$$\mathbf{e}_j = \sigma^{\mathrm{AE}}(\mathbf{W}^{\mathrm{AE}}\mathbf{u}_j + \mathbf{b}_j^{\mathrm{AE}}), \tag{22}$$

where j is the index of neurons and $\sigma^{\mathrm{AE}}, \mathbf{W}^{\mathrm{AE}}, \mathbf{b}^{\mathrm{AE}}$ denote respectively the activation function, the encoder weights and the encoder bias. The vector $\mathbf{e}$ represents the encoded features, related to the input vector $\mathbf{u}$ in the latent space often with a much lower dimension (i.e., $\dim(\mathbf{e}) \ll \dim(\mathbf{u})$). A decoder is then added to approximate the input vector $\mathbf{u}$ via a reconstructed vector $\mathbf{z}$, i.e.,

$$\mathbf{z}_j = \tilde{\sigma}^{\mathrm{AE}}(\tilde{\mathbf{W}}^{\mathrm{AE}}\mathbf{e}_j + \tilde{\mathbf{b}}^{\mathrm{AE}}), \tag{23}$$

with $\tilde{\sigma}^{\mathrm{AE}}, \tilde{\mathbf{W}}^{\mathrm{AE}}, \tilde{\mathbf{b}}^{\mathrm{AE}}$ as the activation function, the decoder weights and the decoder bias. equation (22) (resp. equation (23)) stands for the general formulation of one encoder (resp. decoder) layer while an in-depth AE may consist of several encoder/decoder layers. The AE is then trained with the loss function

$$J\Big([\mathbf{W}^{\mathrm{AE}}, \mathbf{b}^{\mathrm{AE}}, \tilde{\mathbf{W}}^{\mathrm{AE}}, \tilde{\mathbf{b}}^{\mathrm{AE}}]\Big) = \sqrt{\frac{1}{N_{\mathrm{train}}^{\mathrm{AE}}} \sum_{j=1}^{N_{\mathrm{train}}^{\mathrm{AE}}} ||\mathbf{u}_j - \mathbf{z}_j||^2} \tag{24}$$

where $N_{\mathrm{train}}^{\mathrm{AE}}$ denotes the size of the AE training dataset. The AE implemented in this study is a convolutional autoencoder (CAE) which is widely applied to dimension reduction when dealing with multi-dimensional input vectors with structured meshes/cells (e.g., the homogeneous square cells in this study) [4]. Thanks to these convolutional layers, the CAE manages to capture 2D local patterns of fire spread simulations. The exact CAE structures used for different fire events are introduced in section 7 of this paper. For more details about convolutional layers and CNN, interested readers are referred to [62].

5. RNN and LSTM

Since the reduced-order modelling of the state variables is available and the same POD basis/autoencoder can be used for observation data, we aim to establish a surrogate model for predictions in the low-dimensional latent space. With the growing availability of simulation/observation data, there is increasing interest in applying ML approaches to deal with geophysical/atmospheric flow. The recurrent neural networks (RNNs), which address temporal sequences as directed graphs, are of particular interest in handling complex dynamical systems thanks to their ability to capture historical dependencies via internal feedback connections[59]. Among the

different variations of RNNs, the long-short-term-memory (LSTM), first introduced in [43], is adequate of solving long-term dependency problems [13] that classical (also known as "vanilla") RNNs could not achieve. As with other RNNs variants, LSTM has a chain-like structure being composed by repeating the same module shown in Figure 2. In LSTM, instead of only one neural network in each module, four are contained.

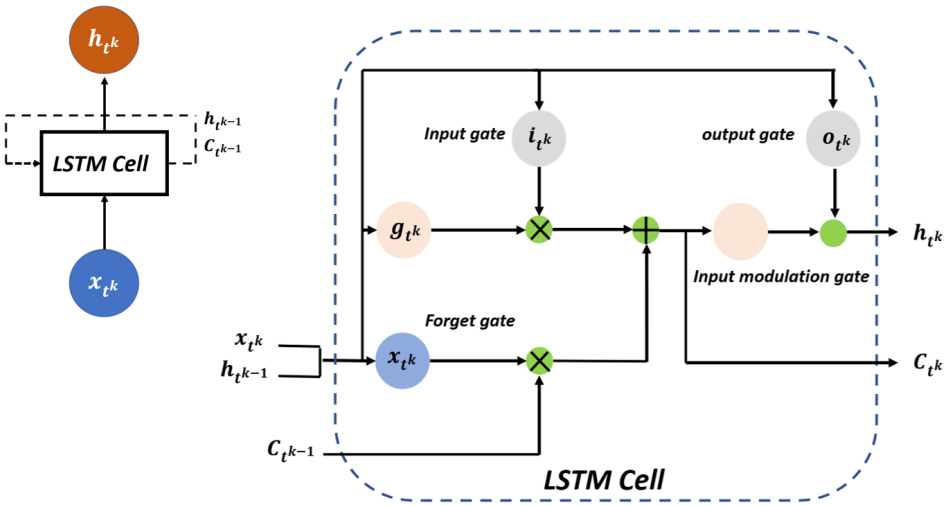


Figure 2: LSTM diagram

The key part of the LSTM network is the cell state $\mathbf{C}_t$ which stores long-term memory information related to historical behaviour. Three gates, each composed of a sigmoid layer (with activate function $\sigma(x) = (1/(1+e^{-x}))$) and a pointwise multiplication operation, are employed to control/adjust the information of the cell state.

- *Forget gate* decides what information is going to be thrown away from the cell state with recurrent variable $\mathbf{h}_{t-1}$ summarising all the information about the past behaviours and $\mathbf{x}_t$ information about the current ones.

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_f) \tag{25}$$

- *Input gate* determines the new information which is going to be added into $\mathbf{C}_{t-1}$ through a *tanh* layer.

$$\tilde{\mathbf{C}}_t = tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_C) \tag{26}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + b_i) \tag{27}$$

where $\tilde{\mathbf{C}}_t$ is multiplied by weight coefficients generated by the input gate, allowing an update of $\mathbf{C}_t$,

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{28}$$

- *Output gate* decides the recurrent output $\mathbf{h}_t$ of the module via previous recurrent output $\mathbf{h}_{t-1}$ and current input information $\mathbf{x}_t$ through a sigmoid layer, i.e.,

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + b_o) \tag{29}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot tanh(\mathbf{C}_t) \tag{30}$$

6. Data-driven Surrogate Model with Latent data assimilation

In this section we propose a surrogate model based on ROM and LSTM with DA described in Algorithm 1. The reduced space is created by the function ϕ in Step 9 of Algorithm 1 by POD following equations (15) and (17) or CAE following equation (22) to (23). The LSTM is trained in the reduced space as shown in Figure 3. To compute the dynamical surrogate model, the trained LSTM is iteratively applied as shown in Step 10 of Algorithm 1. Compared to the traditional many-to-one LSTM network, the sequence-to-sequence structure conduces to accelerate the prediction process and improve the network stability, for instance, by avoiding the vanishing gradient problem [73]. The Assimilation part is also performed in the reduced space as a Latent DA:

$$J_{\text{latent}}(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \hat{\mathbf{B}}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}))^T \hat{\mathbf{R}}^{-1}(\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x})) \tag{31}$$

where $\hat{\mathbf{B}} = \phi(\mathbf{B})$ and $\hat{\mathbf{R}} = \phi(\mathbf{R})$ denote the error covariance matrices in the latent space, $\hat{\mathbf{H}}$ denotes the observation operator in the latent space and where the covariance tuning is improved by the DI01 algorithm in the latent space too.

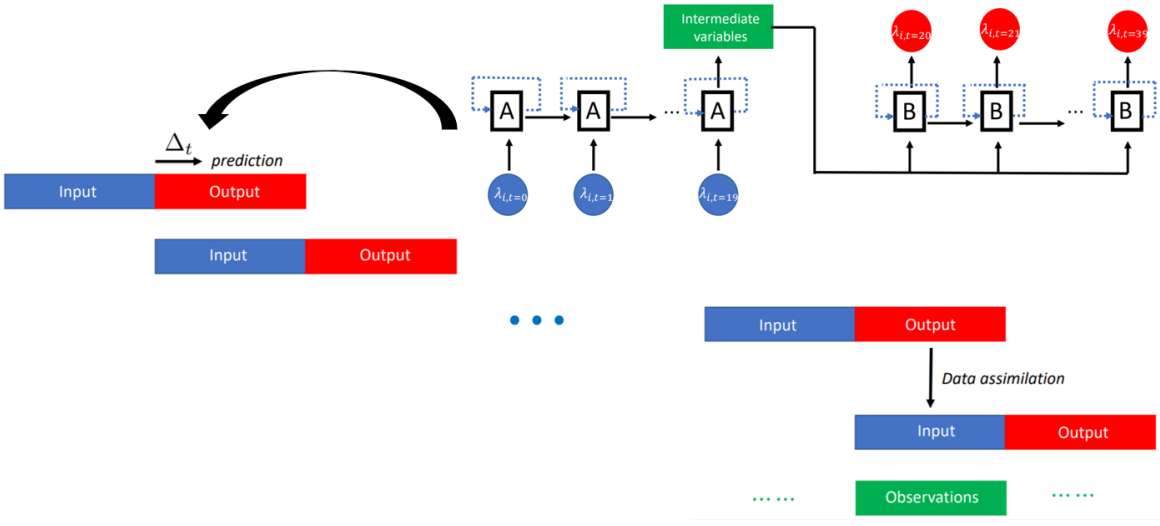


Figure 3: Sequence-to-sequence LSTM predictions of eigenvalues ($i \in \{1, .., 20\}$) with latent data assimilation when observation data are available

The simulation and the observation data are encoded into the same latent space in this modelling, leading to a diagonal structure of the $\hat{\mathbf{H}}$ matrix. The background and the observation error covariances are often set to be diagonal in Latent DA ([4]).

Algorithm 1: combination of ROM, LSTM, DA and DI01

Parameters:

1. Number of time-steps: t_{F}
2. LSTM input length: n_{input}
3. LSTM output length: n_{output}

Inputs:

4. Initial states: $\mathbf{u}_i$, for $i \in \{0..t_{\mathrm{x}}\}$
5. Observation data: $\mathbf{y}_i$, for $i \in T_{\mathrm{y}}$
6. Estimated latent covariance matrices: $\hat{\mathbf{B}}, \hat{\mathbf{R}}$
7. Latent observation operator: $\hat{\mathbf{H}}$
8. Trained LSTM function f^L

Algorithm:

9. projection/encoding $\quad \phi : \begin{cases} \mathbf{u}_i \longrightarrow \mathbf{x}_i^b, i \in \{0..t_{\mathrm{x}}\} \\ \mathbf{y}_i \longrightarrow \hat{\mathbf{y}}_i, i \in T_{\mathrm{y}}. \end{cases} \quad i \longleftarrow t_{\mathrm{x}}$

10. **while** $i < t_F$**:** **do**

$\{\mathbf{x}_{i+1}^b, \mathbf{x}_{i+2}^b, ..., \mathbf{x}_{i+n_{\mathrm{output}}}^b\} = f^L\left(\mathbf{x}_{i-n_{\mathrm{input}}}^b, \mathbf{x}_{i-n_{\mathrm{input}}+1}^b, ..., \mathbf{x}_i^b\right)$

for j *from* $i+1$ *to* $i + n_{output}$ **do**

if $j \in T_y$ **then**

$(\hat{\mathbf{B}}, \hat{\mathbf{R}}) \longleftarrow \mathrm{DI01}(\mathbf{x}_j^b, \hat{\mathbf{y}}_j, \hat{\mathbf{B}}, \hat{\mathbf{R}}, \hat{\mathbf{H}})$

$\mathbf{x}_j^a \longleftarrow \mathrm{DA}(\mathbf{x}_j^b, \hat{\mathbf{y}}_j, \hat{\mathbf{B}}, \hat{\mathbf{R}}, \hat{\mathbf{H}})$

$\mathbf{x}_j^b \longleftarrow \mathbf{x}_j^a$

end

end

$i \longleftarrow i + n_{\mathrm{input}}$

end

outputs: $\{\mathbf{x}_i^b, i \in \{0..t_F\}\}$

As a consequence, $\hat{\mathbf{H}}\hat{\mathbf{B}}\hat{\mathbf{H}}^T$ and $\hat{\mathbf{R}}$ can have the same correlation structure, leading to divergence (i.e., the fixed point of DI01 can be different from the exact error covariances.) of DI01 algorithm as mentioned in [27, 54]. Here we explain in details the mathematical reasons which lead to the divergence of DI01. In order to distinguish the notation of the exact covariances matrices (represent the true estimation error covariances) and the estimated ones, the former is denoted as $\hat{\mathbf{B}}_{\mathrm{E}}, \hat{\mathbf{R}}_{\mathrm{E}}$ while $\hat{\mathbf{B}}_{\mathrm{A}}, \hat{\mathbf{R}}_{\mathrm{A}}$ represent the assumed covariances. In addition, the covariance matrix of the prior innovation vector $\mathbf{d} = \hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_b)$ is denoted by $\mathbf{D}$, i.e., $\mathbf{D} = \mathrm{Cov}(\mathbf{d}, \mathbf{d})$. We remind that the analysis is carried out under the assumption of a linear observation operator $\hat{\mathbf{H}}$, where the explicit expression of $\mathbf{D}$ can be found,

$$\mathbf{D} = \hat{\mathbf{H}}\hat{\mathbf{B}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}. \tag{32}$$

Similar to $\hat{\mathbf{B}}_{\mathrm{A}}, \hat{\mathbf{R}}_{\mathrm{A}}$, the assumed innovation covariance matrix $\mathbf{D}_{\mathrm{A}}$ is defined as

$$\mathbf{D}_{\mathrm{A}} = \hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}_{\mathrm{A}}. \tag{33}$$

Following the assumption of [28], the correlation patterns of these matrices are perfectly identified *a priori*. In other words, for any current iteration q,

$$\hat{\mathbf{B}}_{\mathrm{A},q} = \beta_q \hat{\mathbf{B}}_{\mathrm{E}}, \quad \hat{\mathbf{R}}_{\mathrm{A},q} = \alpha_q \hat{\mathbf{R}}_{\mathrm{E}}. \tag{34}$$

where β_q, α_q are real numbers. Hence, the DI01 algorithm is equivalent to the tuning of two scalar sequences,

$$\beta_{q+1} = s_q^b \beta_q, \quad \alpha_{q+1} = s_q^o \alpha_q. \tag{35}$$

According to the analysis of [54], in the ideal case where equation (34) is satisfied, the multiplicative coefficient updating could be expressed as:

$$\beta_{q+1} = \beta_q \frac{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\mathbf{D}\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A},q}\hat{\mathbf{H}}^T\}}{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A},q}\hat{\mathbf{H}}^T\}}. \tag{36}$$

$$\alpha_{q+1} = \alpha_q \frac{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\mathbf{D}\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{R}}_{\mathrm{A},q}\}}{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{R}}_{\mathrm{A},q}\}}. \tag{37}$$

When $\mathbf{D}_{\mathrm{A},q} = \mathbf{D}$ (i.e. $\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A,q}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}_{\mathrm{A,q}} = \hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}_{\mathrm{E}}$),

$$\frac{\beta_{q+1}}{\beta_q} = \frac{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A},q}\hat{\mathbf{H}}^T\}}{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A},q}\hat{\mathbf{H}}^T\}} = 1, \quad \frac{\alpha_{q+1}}{\alpha_q} = \frac{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{R}}_{\mathrm{A},q}\}}{\mathrm{Tr}\{\mathbf{D}_{\mathrm{A},q}^{-1}\hat{\mathbf{R}}_{\mathrm{A},q}\}} = 1. \tag{38}$$

Therefore both $\{\alpha_q\}$ and $\{\beta_q\}$ converge numerically. If $\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T$ and $\hat{\mathbf{R}}_{\mathrm{E}}$ are of different structures, i.e.,

$$\nexists \tau \in \mathbb{R}, \quad \text{such that} \quad \hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T = \tau \hat{\mathbf{R}}_{\mathrm{E}}, \tag{39}$$

then the convergence of (β_q, α_q) is equivalent to $\hat{\mathbf{B}}_{\mathrm{A}} = \hat{\mathbf{B}}_{\mathrm{E}}$ and $\hat{\mathbf{R}}_{\mathrm{A}} = \hat{\mathbf{R}}_{\mathrm{E}}$ since according to equation (34),

$$\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A,q}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}_{\mathrm{A,q}} = \hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T + \hat{\mathbf{R}}_{\mathrm{E}} \tag{40}$$

$$\iff (\beta_q - 1)\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T = (1 - \alpha_q)\hat{\mathbf{R}}_{\mathrm{E}} \tag{41}$$

$$\iff \beta_q = \alpha_q = 1. \tag{42}$$

On the other hand when $\hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{A}}\hat{\mathbf{H}}^T$ and $\hat{\mathbf{R}}_{\mathrm{A}}$ have the same correlation structure, i.e.,

$$\exists \tau \in \mathbb{R}, \quad \text{such that} \quad \hat{\mathbf{H}}\hat{\mathbf{B}}_{\mathrm{E}}\hat{\mathbf{H}}^T = \tau \hat{\mathbf{R}}_{\mathrm{E}}, \tag{43}$$

fixed-points other than true covariance matrices could be found for DI01 as long as $\frac{(\beta_q - 1)}{(1-\alpha_q)} = \tau$. Therefore, when applying Latent DA with DI01 covariance tuning, choosing a different structure of $\hat{\mathbf{B}}$ and $\hat{\mathbf{R}}$ in the latent space is crucial as specified in section 7 for the wildfire application.

7. Application to wildfire forecasting

7.1. A CA-based simulator

Many studies have attempted to quantify fire progress through an equation governing mathematical model with the help of critical parameters such as fire density or rate of spread (ROS) [60, 67]. As summarized in [56], geophysical features such

as fuel density, landscape slope, and wind speed can severely impact the behaviour of fire progress. Taking these parameters into account, various softwares [32, 2, 42] have been developed to monitoring spatial fire spread processes. A detailed review and comparison of these simulation tools can be found in [56]. In this study, we focus on an operating CA model [2], capable of imitating forest fires with mountainous landscape and complex vegetation distribution, for generating learning data for the LSTM network.

A two-dimensional grid which divides the forest area of study into small square cells with coordinates i, j, as shown in Figure 5(a) is used to simulate the fire progression in all eight possible directions (i.e., $\{i \pm 1, j \pm 1\}/\{i, j\}$) following a burning probability determined by local environmental variables. As stated in [2], the use of regular square cells instead of complex geometry meshes accelerates the fire spread simulation with a lower calculation cost. More precisely, four states (integer numbers) can be assigned to characterize a cell at a discrete time,

- state 1: The cell can not be burned due to lack of forest fuel;
- state 2: The cell has not been ignited ;
- state 3: The cell is burning;
- state 4: The cell has been burned down.

States 1 and 4, either not ignitable or already burned down, will remain invariant during the simulation process. At each discrete time step, the fire propagation to adjacent cells (i.e.,state$(i, j) = 3 \longrightarrow$ state$(i \pm 1, j \pm 1) = 3$) follows a local temporal probability,

$$P_{\text{bun}} = p_h(1 + p_{\text{veg}})(1 + p_{\text{den}})p_s \tag{44}$$

where p_h denotes a standard burning probability, while p_{veg}, p_{den} and p_s depend respectively on the local canopy density, canopy cover and landscape slope of the receiving cell. The transformation equations, which involve, for instance, spread rate modelling and spotting effect, can be found in detail in [2]. The actual values of these physical fields, such as landscape slope, vegetation density, and vegetation cover, are obtained from the Interagency Fuel Treatment Decision Support System (IFTDSS) [1] for corresponding wildfires. An example is given in Figure 5(b,c,d) for the Bear 2017 fire in California. This CA-based operational simulator managed to predict the Spetses fire in 1990 in Greece adequately according to actual observations as shown in [2]. In this work, we have adopted the same optimal values for operational parameters, adjusted through operational experiments, as stated in [2].

[1] https://iftdss.firenet.gov/landing_page/

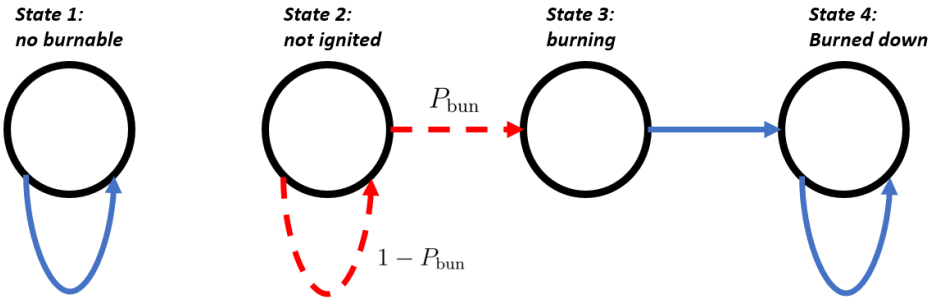
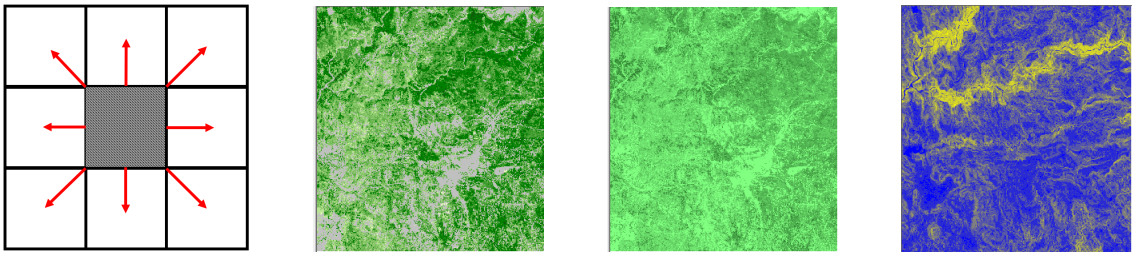


Figure 4: Simple sketch illustrating the state transitions in the CA model when an adjacent cell is burning



(a) (b) (c) (d)

Figure 5: (a): possible directions of fire propagation (b,c,d): respectively the canopy cover, the canopy density and the landscape slope of the study zone which corresponds to the Bear fire in California 2020

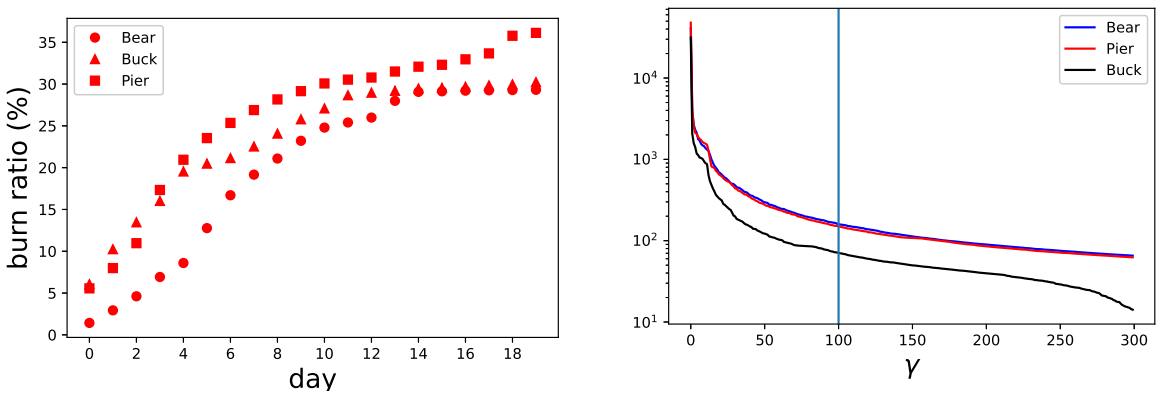
7.2. Study areas

We tested our approach over three recent massive wildfire events in California, including Buck and Pier fires in 2017 and Bear fire in 2020, to ensure the proposed approach is generalizible and robust. Buck fire started on September 2, 2017 and burned 10,450 acres in the Mendocino National Forest around the Napa and Sonoma Valley areas. Pier fire burned 36,556 acres in the Sequoia National Forest, approximately from August 29, 2017 to November 29, 2017. Bear fire is a part of the North Complex fire in the Plumas National Forest in Northern California, started on the morning of August 17, 2020, consisting of numerous lightning fires being managed as one incident. These three fire scenarios are listed among the largest and deadliest wildfires in California since 2017, resulting in an evacuation of hundreds of thousands of local residents.

The issue of evacuation of residents, including transportation and sheltering, is particularly challenging due to the speed and chaotic nature of wildfire spread. Consequently, it also places significant stress on public resources [80]. An accurate near real-time fire forecasting/nowcasting is, thus, crucial to guide effective evacuation plans and on the ground fire suppression efforts. In this study, three different surrogate models are developed for these three large wildfire events in California. Daily fire perimeters derived from satellite thermal observations are used to synchronically adjust the fire spread prediction through data assimilation [69]. Active fires (or hot spots) were detected 4 times a day globally with thermal imagery at

1km from Moderate Resolution Imaging Spectroradiometer (MODIS), aboard the Terra (10:30 and 22:30 local overpass time) and Aqua (13:30 and 01:30 local overpass time) polar orbiting satellites [37]. The Visible Infrared Imaging Radiometer Suite (VIIRS), on board the Suomi National Polar-orbiting Partnership (SNPP) satellite, launched in 2011, further enhanced fire detection capabilities, due to its finer spatial resolution, higher signal-to-noise ratio, dual-gain high-saturation channels, and notable reduction of pixel growth off-nadir [79]. VIIRS provides global coverage of hot spot detections every 12 hours at 13:30 and 1:30 local overpass times [70]. We here combined the level 2 VIIRS I-band active fire product (VNP14IMG) at 375 m with the MODIS standard active fire products (MOD14 and MYD14) and used the natural neighbour geospatial interpolation method to estimate continuous daily fire perimeters [69]. The observational records are extracted from the polygon database to capture daily fire progression for a total of 454 wildfires larger than 200 hectares (500 acres) from January 1, 2012 to December 31, 2020 in California. The MODIS and VIIRS data are available about 2.5 hours after the satellite overpass, allowing for a near real-time fire monitoring.

The exact study area associated with each of the three fire events where the CA simulations take place is shown in Table 1. These square study areas are determined regarding the final progression state observed by satellites. We illustrate the evolution of the burned ratio (i.e., the number of burned cells divided by the number of total cells in the square space) in percentage in Figure 6 (a) for all three fire events. A clear stagnation after 10 to 12 days of the fire start can be observed for each of the three fire scenarios (even though the fire phenomenon may officially last for several weeks), indicating the effects of either fire weather or fire suppression. Since we aim to simulate the early active phase of ferocious fire spread, only the daily evolution of the first 10 to 12 days of those three fires are used to train the LSTM surrogate models. In each case, the observed fire front at the beginning of the fire (i.e., day 0) is considered as a set of ignition points in CA simulations.



(a) (b)

Figure 6: (a): Evolution of the observed burning areas in percentage of the three studied fires; (b): Distribution of eigenvalues in the POD analysis

7.3. ROM, LSTM prediction and DA

Now that the physics-based simulation model and real-time satellite observations are available, we apply the proposed methodology, combining ROM, LSTM and DA

Fire	latitude		longitude		area
	North	South	West	East	
Bear 2020	39.8567	39.7780	−121.1615	−121.0171	$\approx 108\text{km}^2$
Buck 2017	40.2558	40.1707	−123.0791	−122.9734	$\approx 83\text{km}^2$
Pier 2017	36.1909	36.0543	−118.798 698	−118.616 145	$\approx 244\text{km}^2$

Table 1: study areas of the three wildfires

to wildfire predictions. As stated previously in section 4, the objective is to find an optimal tradeoff between complexity and accuracy, that is, computing a lower-dimensional latent space that loses the least possible information. To this end, the performance of POD and CAE with a comparable dimension of the latent space is compared in this section. More precisely, the truncation parameter of the POD approach is fixed as $q = 100$ while the dimension of the latent spaces varies slightly for different fire events. For example, the reduced image is of size 11×11 for the Buck fire as shown in table 2 We first compute the reduced POD basis of the 2D CA fire spread field, consisting of integers $\{1, 2, 3, 4\}$ which represents the current state (either burned, burning or not burned) of each cell. Since the CA simulation employed in this study is stochastic, for each fire event 10 independently simulated fire dynamics, each of 500 or 400 time steps regarding the fire event, are used to compute the vectors of principal components, as described in section 4. After inspecting the distribution of associated eigenvalues as shown in Figure 6(b), the truncation parameter is set to be $\gamma = 100$, where the eigenvalue distributions reach stagnation and the compression accuracy of all three study cases is above 99.8%, as shown in Table 3.

On the other hand, the CAE, trained with the same dataset as POD (i.e., 10 independent CA simulations), consists of four encoding and four decoding layers, as reported in Table 2 where all convolutional layers are set with "padding = same". The number and the placement of the layers has been optimized via a grid search where structures of 3,4,5 layers of encoding/decoding layers have been tested.

A combined ROM- and LSTM-based learning framework is then used to approximate the fire spread dynamics in the latent space for all three fire events. The exact LSTM structure used in all three fire scenario studies is shown in Table 4, while the sequence-to-sequence structure is accomplished thanks to the *RepeatVector* layer which links the output of the previous LSTM layer to each output in the time sequences as shown in Figure 3. The length of both the input and the output time sequence in the LSTM network is set to be 20 CA time steps (equivalent to 12 hours of fire spread time) which is the tradeoff between fast predictions and LSTM stability. The latter is crucial for learning long-term patterns [81]. In fact, since the CA simulations are stochastic at each time step, the LSTM network will be ineffective at directly predicting the fire spread k steps forward when k is too large due to the accumulated randomness. For each fire event, the LSTM training employs 120 independent simulations to form the training and testing dataset by shifting the simulation sequences. For example, if the previous training sample is with $\{0, \ldots, 19\}$ (resp. $\{20, \ldots, 39\}$) time steps as input (resp. output), the next training sample will include time steps $\{1, \ldots, 20\}$ (resp. $\{21, \ldots, 40\}$) as input (resp. output) data.

Layer (type)	Output Shape	Activation
Encoder		
Conv(10×10)	$(648, 633, 1)$	ReLu
MaxPooling(5×5)	$(130, 127, 1)$	
Conv(2×2)	$(130, 127, 1)$	ReLu
MaxPooling(3×3)	$(44, 43, 1)$	
Conv(3×3)	$(44, 43, 1)$	ReLu
MaxPooling(2×2)	$(22, 22, 1)$	
Conv(3×3)	$(22, 22, 1)$	ReLu
MaxPooling(2×2)	$(11, 11, 1)$	
Decoder		
Conv(2×2)	$(11, 11, 1)$	ReLu
MaxPooling(2×2)	$(22, 22, 1)$	
Conv(3×3)	$(22, 22, 1)$	ReLu
MaxPooling(5×5)	$(330, 330, 1)$	
Conv(2×2)	$(330, 330, 1)$	ReLu
MaxPooling(2×2)	$(660, 660, 1)$	
Cropping (12,27)	$(648, 633, 1)$	
Conv(10×10)	$(648, 633, 1)$	Sigmoid

Table 2: LSTM structure for learning physics-based CA simulations

Fire	Bear 2020	Buck 2017	Pier 2017
C_a	99.8%	99.9%	99.9%
C_r	0.017%	0.024%	0.014%

Table 3: Compression accuracy and rate with truncation parameter $\gamma = 100$ for different fire events

Layer (type)	Output Shape	Activation
LSTM	(200)	ReLu
RepeatVector	$(20, 200)$	
LSTM	$(20, 100)$	ReLu
Dense	$(20, 100)$	Linear
Dense	$(20, 100)$	Linear
Time distributed	$(20, 100)$	Linear

Table 4: LSTM structure for learning physics-based CA simulations

After acquiring the LSTM model in the latent space, real-time satellite observations are used to regularize the fire prediction according to current circumstances. The post-processed observations, indicating the current burned area, are first projected to the same latent space determined by $\boldsymbol{Q}_\gamma$ as state variables, with the same truncation parameter $\gamma = 100$. Since the observations are obtained on a daily basis, the same observation is used to assimilate 20 consecutive prediction steps to ensure a continuous prediction. This is crucial for the following LSTM forecasting. For each assimilation window, the ratio between the background and the observation information (i.e., $\mathrm{Tr}(\hat{\mathbf{B}})$ and $\mathrm{Tr}(\hat{\mathbf{R}})$) is tuned via the DI01 approach for a fixed number of 5 iterations. The combination of ROM, LSTM, DA and covariance tuning, as described above, is summarized in Algorithm 1.

8. Results

In this section, the results of the proposed algorithm scheme which combines ROM, LSTM and DA are discussed for three recent fire events in California introduced in section 7.

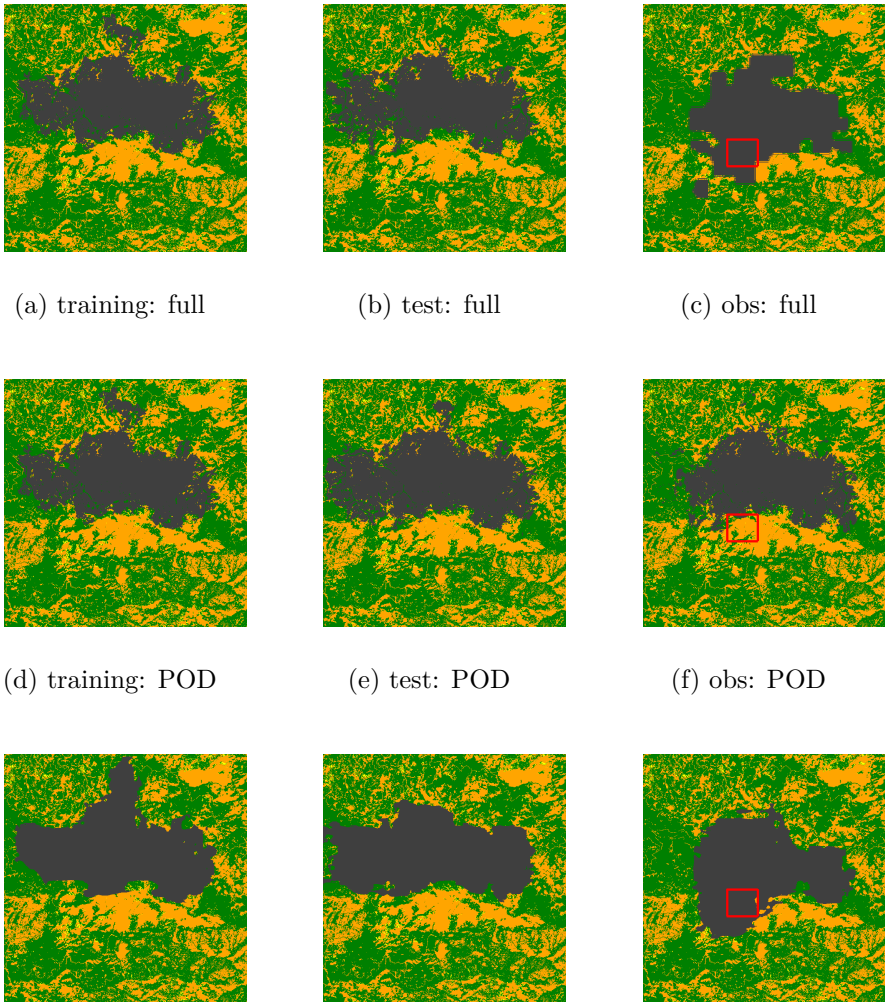
8.1. ROM: POD vs. AE

We now focus on the construction of the latent space based on POD or AE compression strategies. The CAE has been tested for the three fire events with the same layer structures (as shown in Table 2), except the cropping layer that adjusts the output dimension to the same as input.

The original and reconstructed burned areas respectively from the training dataset, the testing dataset (CA simulations not used for training/compression) and the satellite observations, are shown in Figure 7 for the Buck 2017 fire as an example. The POD latent space is of dimension 100 while the CAE latent space is of dimension 11×11 as reported in Table 2. Observing the first two columns of Figure 7, the POD approach overperforms the CAE in terms of reconstruction accuracy for both the training and test dataset (generated via CA simulations). However, the CAE exhibits significantly better flexibility when dealing with satellite observation data for reconstructing burned areas that haven't been seen in the training dataset (for example, the red square in Figure 7(c,f,i)). These findings are consistent with the observations in Figure 8, which illustrates the evolution of the relative root mean square error (R-RMSE) defined as:

$$\epsilon_{\text{R-RMSE},(t)}(\iota) = \sqrt{\sum_{j=1}^{\iota} \frac{1}{\iota} \Big(\frac{||\mathbf{u}_t^{\text{original},j} - \hat{\mathbf{u}}_t^j||}{||\mathbf{u}_t^{\text{original},j}||} \Big)^2}, \tag{45}$$

where $\hat{\mathbf{u}} \in \{\mathbf{u}^{\text{POD}}, \mathbf{u}^{\text{CAE}}\}$ is the reconstructed fire spread field and ι denotes the ensemble size where R-RMSE is evaluated. Here, $\iota = 10$ for the training and the testing dataset of CA simulations while $\iota = 1$ for satellite observations. Since the LSTM surrogate model is trained using purely CA simulations while the satellite observations are only used for DA corrections, we choose to proceed with the POD approach for ROM in this study.

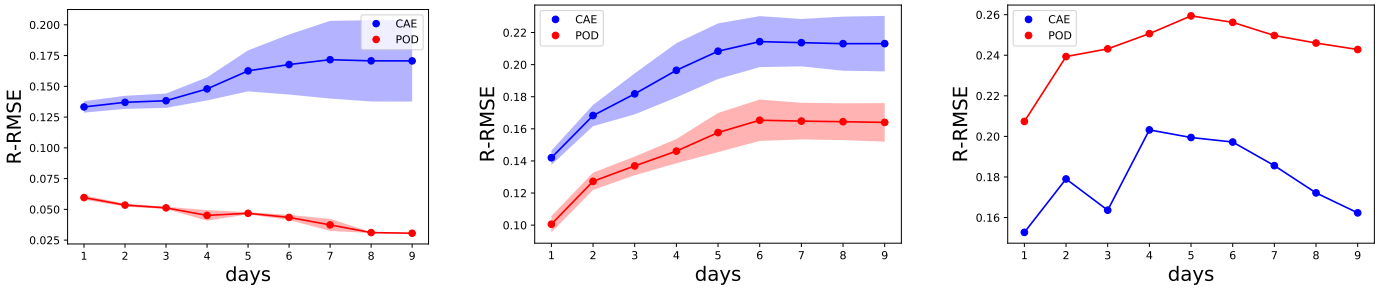


(a) training: full (b) test: full (c) obs: full

(d) training: POD (e) test: POD (f) obs: POD

(g) training: AE (h) test: AE (i) obs: AE

Figure 7: Comparison of POD and CAE respectively in the training (CA simulations), testing (CA simulations) and observation dataset



(a) training (b) testing (c) observations

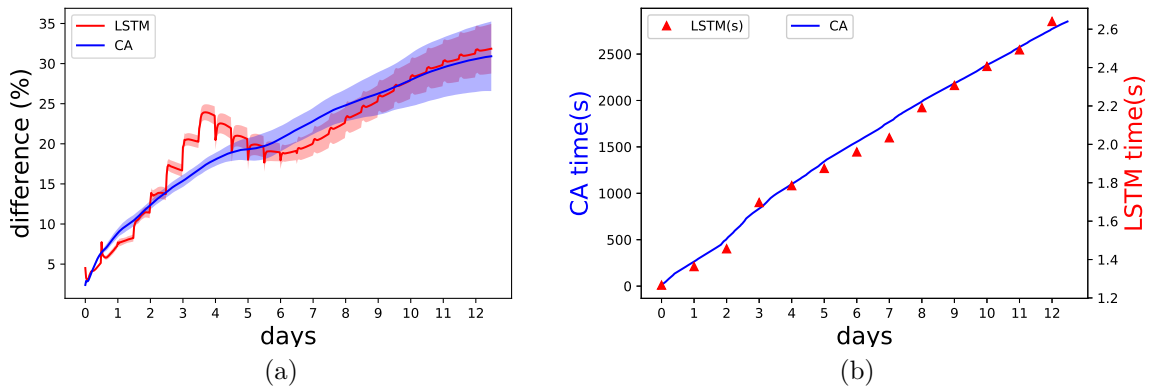
Figure 8: R-RMSE error for POD and CAE in the training/testing/observation dataset where the transparent zones represent the std

8.2. LSTM in the latent space

We first investigate the performance of the LSTM surrogate model, trained on 120 CA simulations for each of the three fire scenarios. For the sake of brevity, we

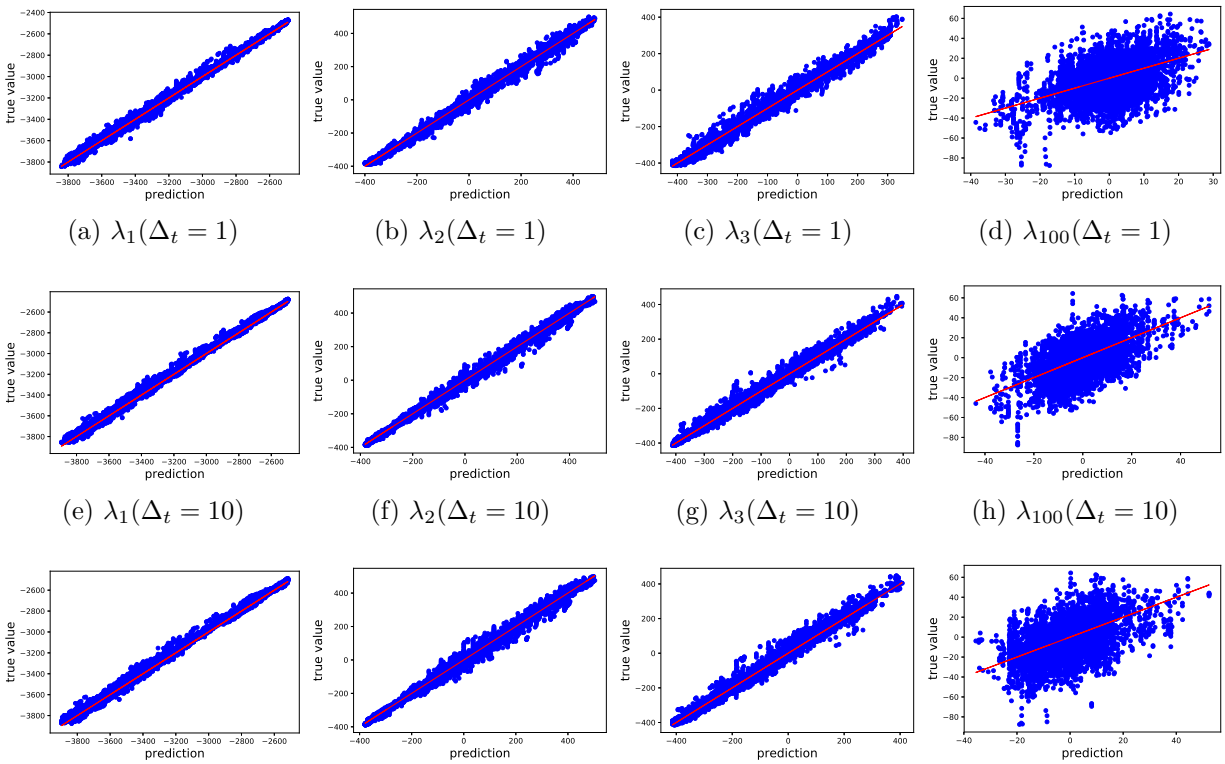
illustrate and analyse the results obtained for the Bear fire in this section as the outcome of two other fire events can be found in the appendix with similar conclusions. The prediction results against the CA simulations (blue circles), both in the latent space, for the testing dataset is shown in Figure 10 where we illustrate the outcome of the first three (i.e., $\lambda_{\{1,2,3\}}$ which contain most representative components) together with the 100^{th} eigenvalues λ_{100} for different prediction length Δ_t. What stands out in Figure 10 is the high precision of predictions for $\lambda_{\{1,2,3\}}$ regardless the prediction length, which confirms the choice of $k = 20$ in the sequence-to-sequence LSTM model setup. In other words, the ML model manages to handle the diffusion randomness of 20 CA steps and the prediction results do not deviate. As for λ_{100}, since the 100^{th} eigenvalues in the POD approach contain a considerable amount of noise, it is to be expected that the LSTM prediction is cumbersome to fit the exact target values. We now represent the reconstructed prediction results in the full physical space following equation (19). Since the fire spread fields consist of integer numbers, an integer round-off filter is added for postprocessing. This integer filter is only required for comparison, which is not necessary for intermediate LSTM steps.

We display in Figure 14(a) the averaged normalized difference and associated standard deviation (std) (shown by the transparent zones) for the LSTM-CA and CA-CA comparison, both evaluated for 10 different simulations, for the 2020 Bear fire in the full space. More precisely, both LSTM and CA initiate with the burned area at day 0 (determined by the satellite observation) where the LSTM observes the first 20 diffusion steps (of the corresponding CA simulation) in the latent space to launch the surrogate predictions. It must be emphasised that due to the random nature of the CA approach, independent simulations can lead to different fire spread scenarios. It can be stated that the LSTM surrogate model which is trained using more than 100 independent simulations, exhibits similar performance to a full CA model (consistent with the findings in Figure 10) with considerably less computational time, as illustrated in Figure 14(b). More precisely, a CA simulation costs in average about 45 minutes using a laptop CPU, while an online evaluation of LSTM model costs less than three seconds.



(a) (b)

Figure 9: (a): Averaged R-RMSE difference of CA-LSTM and CA-CA of the Bear fire; (b): Comparison of the computational time (in seconds) of CA (left axis) and LSTM (right axis)



(a) $\lambda_1(\Delta_t = 1)$ (b) $\lambda_2(\Delta_t = 1)$ (c) $\lambda_3(\Delta_t = 1)$ (d) $\lambda_{100}(\Delta_t = 1)$

(e) $\lambda_1(\Delta_t = 10)$ (f) $\lambda_2(\Delta_t = 10)$ (g) $\lambda_3(\Delta_t = 10)$ (h) $\lambda_{100}(\Delta_t = 10)$

(i) $\lambda_1(\Delta_t = 20)$ (j) $\lambda_2(\Delta_t = 20)$ (k) $\lambda_3(\Delta_t = 20)$ (l) $\lambda_{100}(\Delta_t = 20)$

Figure 10: Eigenvalues predictions in the latent space by LSTM with different prediction length of Bear 2020 fire event

8.3. Latent DA

Once the LSTM surrogate model is trained, we make use of satellite observations to adjust the so-obtained burned area predictions. To quantify the difference between observed and predicted/assimilated fire spread fields (respectively denoted as $\mathbf{u}^{\text{obs}}$ and $\hat{\mathbf{u}} \in \{\mathbf{u}^{\text{LSTM}}, \mathbf{u}^{\text{DA}}, \mathbf{u}^{\text{LSTM+DA}}\}$), the relative root mean square error (R-RMSE) is defined as:

$$\epsilon_{\text{R-RMSE},(t)}(\iota) = \sqrt{\sum_{j=1}^{\iota} \frac{1}{\iota} \left(\frac{||\mathbf{u}_t^{\text{obs},j} - \hat{\mathbf{u}}_t^j||}{||\mathbf{u}_t^{\text{obs},j}||} \right)^2}. \tag{46}$$

ι denotes the size of the ensemble of trajectories where R-RMSE is estimated. $\mathbf{u}^{\text{LSTM+DA}}$ represents the current LSTM prediction in the full space which benefits DA corrections from previous time steps. As described in section 7, the LSTM model exploits a sequence-to-sequence structure with $k = 20$, which is equivalent to about 12 hours in real-time. The LSTM predictions are then assimilated using daily satellite observations through DA techniques. The background matrix, which typifies prior error covariance in the latent space, is set to be diagonal (i.e., $\hat{\mathbf{B}} = \mathbf{I}_{100}$) because the POD principal components are uncorrelated. Considering that the DI01 approach exhibits better performance when the correlation patterns of $\hat{\mathbf{B}}$ and $\hat{\mathbf{R}}$ are significantly different [28], the observation error covariance here is set to be isotropic, following the second-order auto-aggressive (SOAR) function,

$$\varphi_{\mathbf{R}}(r) = \left(1 + \frac{r}{L_{\hat{\mathbf{R}}}}\right) \exp(-\frac{r}{L_{\hat{\mathbf{R}}}}), \tag{47}$$

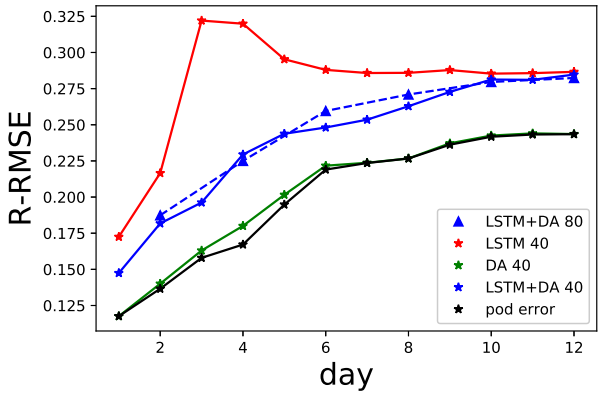
where $r(i,j) = |i-j|, \forall \{i,j\} \in \{1,...,100\}^2$ denote the distance in the latent space of dimension $\gamma = 100$ and $L_{\hat{\mathbf{R}}}$ is the correlation scale length, considering as an hyperparmater in this study. According to numerical experiments, the value is fixed as $L_{\hat{\mathbf{R}}} = 15$ in this study. The so defined $\hat{\mathbf{B}}$ and $\hat{\mathbf{R}}$ matrices are used as initial values in the DI01 covariance tuning. In general, specifying error covariances in the latent space of dynamical systems remains an open question and further research can be considered to address this question.

The evolution of averaged R-RMSE related to 10 CA simulations for the Bear 2020, Buck 2017 and Pier 2017 wildfires are illustrated in Figure 11. In fact, for distinct CA simulations, only the first 20 simulation steps used as initial inputs to launch the LSTM surrogate model are different. Thus, the variance of the estimated R-RMSE is insignificant compared to its absolute value. As shown by the red curve, the difference between LSTM predictions and satellite observations peaks near the end of the first week for all three fire events where the burned areas grow most rapidly as shown in Figure 6(a). Around one-third of burned areas haven't been successfully predicted. This difference can be significantly reduced by performing DA techniques in the latent space, as shown by the green curves, with an R-RMSE comparable to the POD accuracy indicated in Table 3. More importantly, the DA correction also improves the LSTM accuracy substantially for future predictions as illustrated by the blue curves. The solid blue line depicts the prediction with a day-to-day DA correction, while the dashed blue line represents the case where DA takes place every two days. For both cases, the R-RMSE curves are considerably lower than pure LSTM simulations (red), being relatively close to DA corrections (green). Sub-figures (b,d,f) in figure 11 demonstrate the strength of online covariance tuning. In all three study cases, DA and assimilated LSTM with covariance tuning (i.e., "DA cov" and "DA+LSTM cov") result in significantly lower prediction error compared to DA and assimilated LSTM without covariance tuning (i.e., "DA no cov" and "DA+LSTM no cov"). Based on previous analysis, it could be concluded that the latent DA contributes to a long-term impact of improvement in the forecasting accuracy, which is challenging when combining lower-dimensional surrogate models with real-time observation data.

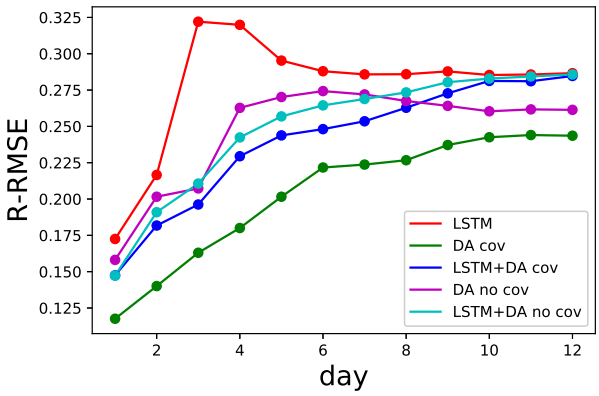
As an example, the exact prediction and assimilation results in the full space from day 1 to day 5 of the Buck 2017 fire are displayed in Figure 12 where the image background consists of the distribution of the vegetation(canopy) cover. The LSTM-observation difference can be noticed right from day 1 (c.f., Figure 12 (a) and (b)), with an increasing tendency until day 5. On the other side, the assimilated states, after being decoded to the full physical space, succeed in combining model predictions and satellite measures as shown in the last column of Figure 12. The model forecastings (g,k,o,s) are respectively based on previous DA corrections (d,h,i,p). Consistent with the analysis of Figure 11, the outcome of LSTM + DA is much closer to the observations of the next time step, in comparison with pure LSTM predictions. The comparison between "obs" and "DA"/"DA + LSTM" columns in Figure 11 also reveals the fact that some areas (e.g., red subdomain) can be more difficult to regulate compared to others (e.g., blue subdomains). This fact is linked to the limitation of ROM and CA simulations. Since the CA simulations depend heavily on the distribution of vegetation, the zone framed by the red square has a very low probability to be burned, resulting in zero variance associated with the corresponding area when computing the covariance matrix in POD analysis. There-

fore, the latent space (i.e., the principal components) contains no information of those cells in the square, leading to an imperfect reconstruction of the satellite observations in the full space. In general, it stands for a major bottleneck for all data compression methods when a significant difference exists between the training (CA simulations in this study) and actual data (satellite observations in this study).

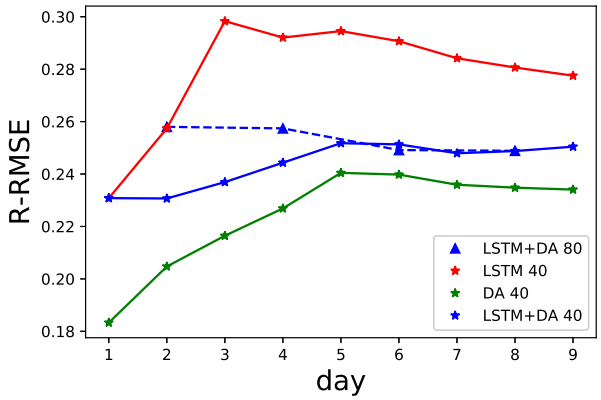
The averaged online computational time of different approaches, including CA, LSTM and DA, is reported in Table 5 for the three fire events. Despite the offline training of AE and LSTM necessitates the use of a high-performance computer (HPC), the online computation is carried out on a laptop CPU as shown in Table 5. More precisely, the offline training of AE and LSTM is performed in the HPC system of Imperial College, and the computing node harnessed one NVIDIA GPU P1000 (RAM 96GB) within 2 hours of computational time for each fire event while the online evaluation takes only several seconds for each time step as shown in section 5. For comparison purposes, the approximately estimated computational time of Flammap software [32], which is commonly used for operational fire management, on the corresponding study areas for one CA step is also shown in the last column of Table 5. For each of the three fire events, the LSTM prediction is at least 1000 times faster in average compared to Flammap or CA simulations. In addition, the efficiency of latent DA and DI01, which are often considered to be computationally expensive and time-consuming in industrial applications, is also demonstrated in Table 5. As a matter of fact, in average each latent covariance tuning process is at least 50 times faster than a step of CA/Flammap simulation. On the other hand, implementing DA in the full physical space with non-diagonal covariance matrices is extremely challenging for a laptop CPU, if not infeasible. In summary, the LSTM-based surrogate model coupled with latent DA proposed in this study, exhibits great efficiency and high accuracy for real-time fire nowcasting/forecasting.



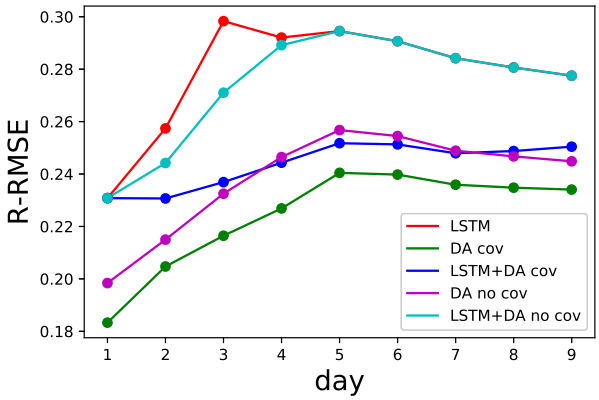
(a) Bear 2020: latent LSTM and DA



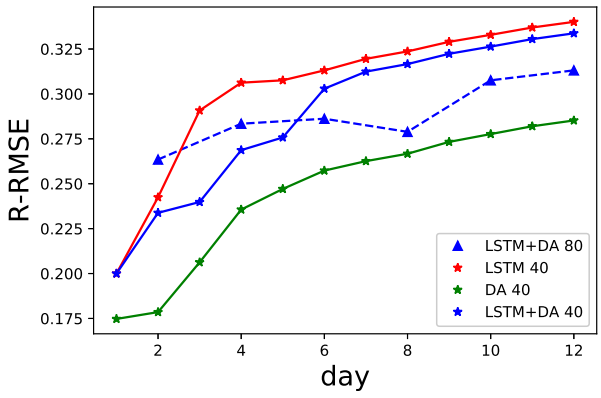
(b) Bear 2020: effect of the covariance tuning



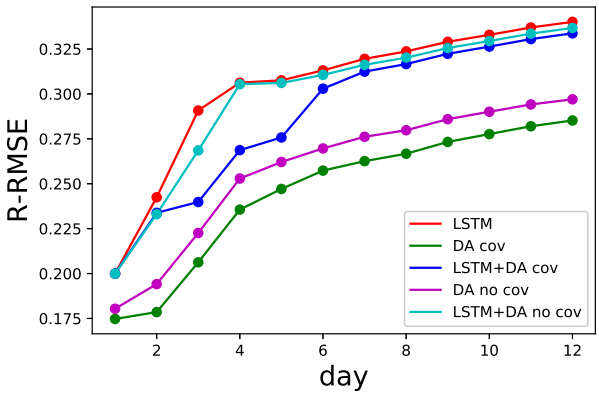
(c) Buck 2017: latent LSTM and DA



(d) Buck 2017: effect of the covariance tuning



(e) Pier 2017: latent LSTM and DA

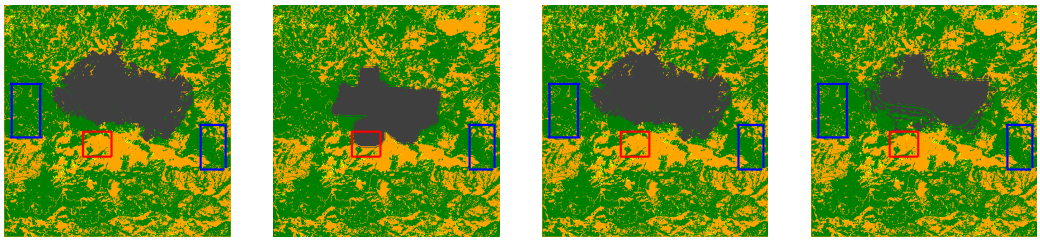


(f) Pier 2017: effect of the covariance tuning

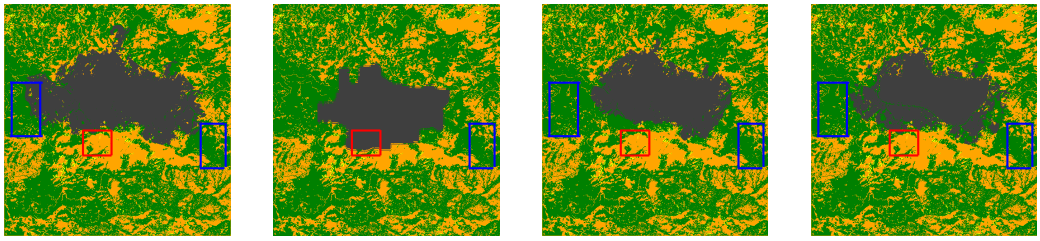
Figure 11: Evolution of R-RMSE for different approaches where LSTM+DA represents the predictions based on the assimilated results of previous time steps

Fire	simulation	LSTM	DA	DI01	Flammap
Bear	$3.85s$	$3.1e^{-3}s$	$5.8e^{-3}s$	$8.7e^{-2}s$	$10 \sim 30$s
Buck	$2.96s$	$5.26e^{-3}s$	$4.26e^{-3}s$	$2.12e^{-2}s$	$5 \sim 20$s
Pier	$8.28s$	$6.28e^{-3}s$	$5.36e^{-3}s$	$2.68e^{-2}s$	$10 \sim 30$s

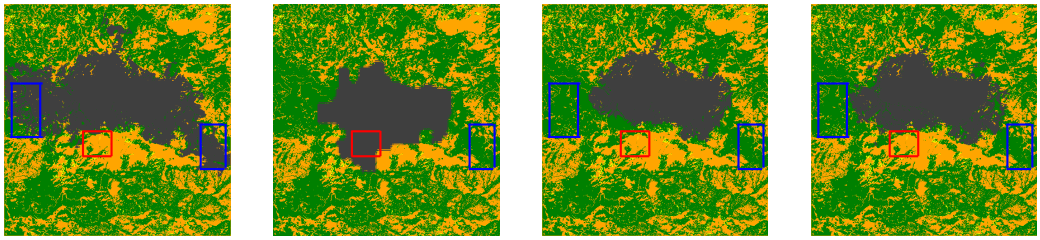
Table 5: Averaged computational time for one time-step using different approaches



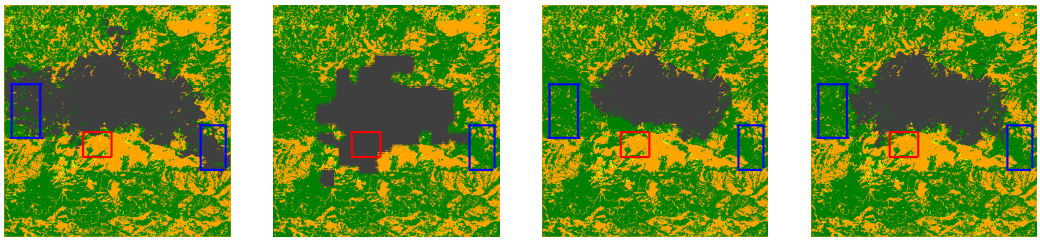
(a) LSTM(day 1) (b) obs(day 1) (c) LSTM+DA(day 1) (d) DA(day 1)



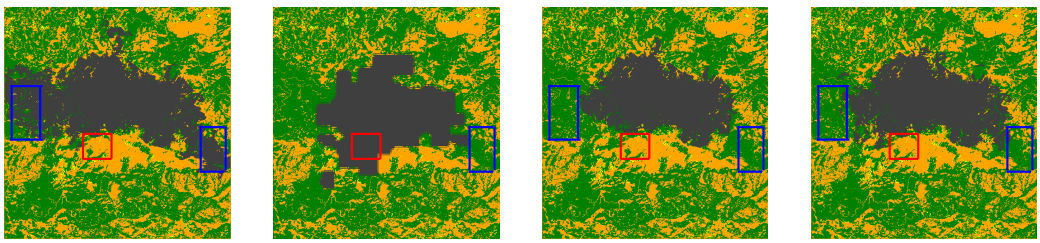
(e) LSTM(day 2) (f) obs(day 2) (g) LSTM+DA(day 2) (h) DA(day 2)



(i) LSTM(day 3) (j) obs(day 3) (k) LSTM+DA(day 3) (l) DA(day 3)



(m) LSTM(day 4) (n) obs(day 4) (o) LSTM+DA(day 4) (p) DA(day 4)



(q) LSTM(day 5) (r) obs(day 5) (s) LSTM+DA(day 5) (t) DA(day 5)

Figure 12: Buck fire: Comparison of (i) pure machine learning prediction (1st column); (ii) preprocessed satellite observations (2nd column); (iii) machine learning prediction with DA (3rd column) and (iv) DA reconstruction (4th column)

9. Conclusions and future work

A major challenge for wildfire prediction models lies in the complexity of the physics-based simulations, leading to difficulties for implementing operational real-time fire forecasting. In this work, we propose a new algorithm scheme that combines physical simulations, reduced order modelling, recurrent neural networks and data assimilation in consecutive order. We first compute the latent space by capturing the governing dynamics using a much smaller number of variables compared to the full physical/geological space. The uses of POD and CAE have been discussed and compared with the observed spread of realistic fire examples. The computational cost is shown significantly reduced for both LSTM surrogate model training and near real-time data assimilation in the latent space. In addition, the use of stochastic CA simulations in ROM and LSTM in this study improves the flexibility of the trained data-driven model, which is crucial when implementing real-time DA. Three recent wildfires in California are investigated with historical satellite images considered as observations in latent DA. We have shown that, with error covariance tuning algorithms, data assimilation manages to provoke long-term impact on enhancing the model prediction accuracy in all three fire events. The online evaluation of the surrogate model is considerably faster than running a full wildfire simulation (either CFD or CA). The framework proposed in this paper can be applied to other spatial temporal dynamics such as air pollution monitoring or environmental epidemiology.

This study is a first attempt to improve the efficiency of fire forecasting while maintaining sufficient accuracy, by using deep learning to reduce the order of complex process-based models and using DA to constrain and adjust near real-time prediction. As a proof of concept, we used simulations from a simpler version of fire spread module (CA) as one example of process-based models. The CA model does not take into account quite a few complex processes of fire behaviours such as fire-generated weather, surface-to-crown fire transition [77], fire spotting. One application is to predict the efficacy of the on the ground fire suppression strategies such as placing fuel breaks and dropping fire retardants. This can be done by modifying the existing fuel maps to represent the impact of suppression on fuels. A full investigation calls for additional work and a larger project, which is beyond the scope of this initial attempt. For example, we can build the surrogate model from some other fire behavior models such as FARSITE, FLAMMAP [32], SPARK [41], and BehavePlus [5] to simulate at various complexity. Future work will benefit from learning from these simulations, augmented with historical fire observations, to add probability of surface-to-crown transition and spotting spread in our framework. From previous works in air pollution [61], we have seen that when more realistic models with more physical modellings involved are available, it can also significantly improve the performance of the ML approach

Further study is also needed to test the machine learning from more complex and realistic wildfire simulations, that consider the probability of surface-to-crown transition and spotting spread in our framework [77, 25]. Also note that in this pilot study, individual ROMs and LSTM surrogate models are trained for different fire events, which can be computationally expensive for offline simulation and training. An important next step is to develop ecoregion specific surrogate models by learning from a group of diverse wildfires. This will reduce the computational cost for training

and improve generalizability for operational real world forecasting.

Acknowledgements

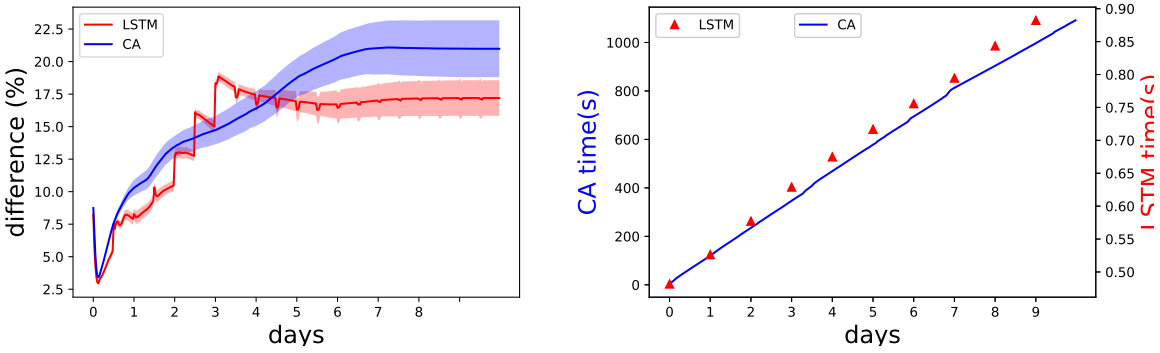
This research is funded by the Leverhulme Centre for Wildfires, Environment and Society through the Leverhulme Trust, grant number RC-2018-023. This work is partially supported by the EP/T000414/1 PREdictive Modelling with QuantIfication of UncERtainty for MultiphasE Systems (PREMIERE).

Abbreviations

BLUE	best linear unbiased estimator
DI01	Desroziers & Ivanov tuning method
3D-Var	3D Variational
ML	machine learning
DL	deep learning
DA	data assimilation
NN	neural network
RNN	recurrent neural network
CNN	convolutional neural network
SVM	support vector machine
LSTM	long short-term memory
POD	proper orthogonal decomposition
CA	cellular automata
GA	genetic algorithms
NWP	numerical weather prediction
CFD	computational fluid mechanics
ROM	reduced-order modelling
1D	one-dimensional
2D	two-dimensional
R-RMSE	relative root mean square error
AE	AutoEncoder
CAE	Convolutional AutoEncoder
HPC	high-performance computer
std	standard deviation
VIIRS	Visible Infrared Imaging Radiometer Suite
MODIS	Moderate Resolution Imaging Spectroradiometer
SNPP	Suomi National Polar-orbiting Partnership

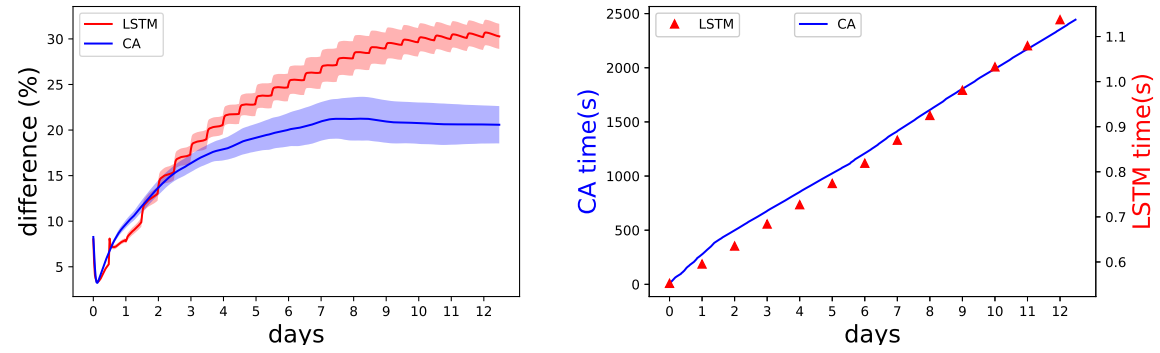
Appendix

CA-based LSTM surrogate model



(a) (b)

Figure 13: (a): Averaged R-RMSE difference of CA-LSTM and CA-CA of the Buck fire; (b): Comparison of the computational time (in seconds) of CA (left axis) and LSTM (right axis)



(a) (b)

Figure 14: (a): Averaged R-RMSE difference of CA-LSTM and CA-CA of the Pier fire; (b): Comparison of the computational time (in seconds) of CA (left axis) and LSTM (right axis)

Maximum Likelihood property of DI01

In this section, we prove that each iteration of DI01 is equivalent to a Maximum Likelihood tuning of the error amplitudes (i.e., Tr($\mathbf{B}$) and Tr($\mathbf{R}$)). In fact, as stated by [17], DI01 is equivalent to tune a Maximum Likelihood algorithm to the innovation covariances matrix $\mathbf{D}(s = \{s^b, s^o\})$ (as defined in equation (32)) parameterized by the multiplicative coefficients. The proof is synthesised based on the work of [54], [17] and [28] with some simplifications.

At each DI01 iteration,

$$\mathbf{D}_{q+1}(s_q^b, s_q^o) = \mathbf{H} s_q^b \mathbf{B}_q \mathbf{H}^T + s_q^o \mathbf{R}_q. \tag{48}$$

For the sake of simplicity, set $\mathbf{D} = \mathbf{D}_{q+1}, \mathbf{d} = \mathbf{d}_{q+1}, \mathbf{B} = \mathbf{B}_q, \mathbf{R} = \mathbf{R}_q, s^b = s_q^b, s^o = s_q^o$ in this section. The conditional probability density function of the innovation quantity $\mathbf{d}$ could be written as:

$$f(\mathbf{d}|\mathbf{s}) = \frac{1}{\sqrt{(2\pi)^p \det(\mathbf{D}(\mathbf{s}))}} exp\Big(-\frac{1}{2}\mathbf{d}^T\mathbf{D}(\mathbf{s})^{-1}\mathbf{d}\Big) \tag{49}$$

where p denotes the dimension of $\mathbf{d}$. We then deduce the log-likelihood function,

$$\mathcal{L}(\mathbf{s}) = -\log(f(\mathbf{d}|\mathbf{s})) = \frac{p}{2}\log(2\pi) + \frac{1}{2}\log[\det(\mathbf{D}(\mathbf{s}))] + \frac{1}{2}\mathbf{d}^T D(\mathbf{s})^{-1}\mathbf{d}. \tag{50}$$

The minimum of this function should satisfy

$$\text{For all component } \ s \ \text{ of } \ \mathbf{s}, \frac{\partial \log[\det(\mathbf{D}(\mathbf{s}))]}{\partial s} + \mathbf{d}^T\frac{\mathbf{D}(\mathbf{s})^{-1}}{\partial s}\mathbf{d} = 0. \tag{51}$$

Using the following algebraic properties,

$$\log[\det(\mathbf{D}(\mathbf{s}))] = \mathrm{Tr}[\log\mathbf{D}(\mathbf{s})], \tag{52}$$

$$\frac{\partial \mathrm{Tr}[\log(\mathbf{D}(\mathbf{s}))]}{\partial s} = \mathrm{Tr}\Big[\mathbf{D}(\mathbf{s})^{-1}\frac{\partial \mathbf{D}(\mathbf{s})}{\partial s}\Big], \tag{53}$$

$$\frac{\mathbf{D}(\mathbf{s})^{-1}}{\partial s} = -\mathbf{D}(\mathbf{s})^{-1}\frac{\mathbf{D}(\mathbf{s})}{\partial s}\mathbf{D}(\mathbf{s})^{-1}, \tag{54}$$

equation (51) could be simplified. For the first term on the left side, according to equation (52) and (53),

$$\frac{\partial \log[\det(\mathbf{D}(\mathbf{s}))]}{\partial s} = \frac{\mathrm{Tr}[\partial \log\mathbf{D}(\mathbf{s})]}{\partial s} = \mathrm{Tr}\Big[\mathbf{D}(\mathbf{s})^{-1}\frac{\partial \mathbf{D}(\mathbf{s})}{\partial s}\Big] = \mathrm{Tr}\Big[\frac{\partial \mathbf{D}(\mathbf{s})}{\partial s}\mathbf{D}(\mathbf{s})^{-1}\Big]. \tag{55}$$

In the case of DI01, $\mathbf{s} = \{s^b, s^o\}$ therefore

$$\frac{\partial \mathbf{D}(\mathbf{s})}{\partial s^b} = \mathbf{H}\mathbf{B}\mathbf{H}^T, \quad \frac{\partial \mathbf{D}(\mathbf{s})}{\partial s^o} = \mathbf{R}. \tag{56}$$

As for the second term of equation (51), according to equation (54),

$$\mathbf{d}^T\frac{\mathbf{D}(\mathbf{s})^{-1}}{\partial s^b}\mathbf{d} = -\mathbf{d}^T\mathbf{D}(\mathbf{s})^{-1}\frac{\mathbf{D}(\mathbf{s})}{\partial s^b}\mathbf{D}(\mathbf{s})^{-1}\mathbf{d} = -\mathbf{d}^T\mathbf{D}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}\mathbf{d} \tag{57}$$

$$\mathbf{d}^T\frac{\mathbf{D}(\mathbf{s})^{-1}}{\partial s^o}\mathbf{d} = -\mathbf{d}^T\mathbf{D}(\mathbf{s})^{-1}\frac{\mathbf{D}(\mathbf{s})}{\partial s^o}\mathbf{D}(\mathbf{s})^{-1}\mathbf{d} = -\mathbf{d}^T\mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}\mathbf{d} \tag{58}$$

equation(51) is then equivalent to the following two marginal formulas,

$$\mathrm{Tr}[\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}] - \mathbf{d}^T\mathbf{D}^{-1}\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}\mathbf{d} = 0 \tag{59}$$

$$\mathrm{Tr}[\mathbf{R}\mathbf{D}^{-1}] - \mathbf{d}^T\mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}\mathbf{d} = 0. \tag{60}$$

On the other hand, from equation (13), one can deduce

$$2J_b(\mathbf{x}_a) = s^b\mathrm{Tr}[\mathbf{K}\mathbf{H}] \tag{61}$$

$$||\mathbf{K}(\mathbf{y} - \mathbf{H}x_b)||_{\mathbf{B}^{-1}} = s^b\mathrm{Tr}[\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}\mathbf{H}] \tag{62}$$

$$(\mathbf{K}\mathbf{d})^T(s^b\mathbf{B})^{-1}\mathbf{K}\mathbf{d} = s^b\mathrm{Tr}[\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}] \tag{63}$$

$$\mathbf{d}^T\mathbf{D}^{-1}\mathbf{H}(s^b\mathbf{B})^T(s^b\mathbf{B})^{-1}(s^b\mathbf{B})\mathbf{H}^T\mathbf{D}^{-1}\mathbf{d} = s^b\mathrm{Tr}[\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}] \tag{64}$$

$$\mathbf{d}^T\mathbf{D}^{-1}\mathbf{H}\mathbf{B}^T\mathbf{B}^{-1}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}\mathbf{d} = \mathrm{Tr}[\mathbf{H}\mathbf{B}\mathbf{H}^T\mathbf{D}^{-1}]. \tag{65}$$

Meanwhile,

$$2J_o(\mathbf{x}_a) = s^o \mathrm{Tr}[\mathbf{I} - \mathbf{KH}] \tag{66}$$

$$((\mathbf{I} - \mathbf{HK})\mathbf{d})^T \mathbf{R}^{-1}((\mathbf{I} - \mathbf{HK})\mathbf{d}) = s^o \mathrm{Tr}[\mathbf{I} - \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}] \tag{67}$$

$$\mathbf{d}^T(s^o\mathbf{R}\mathbf{D}^{-1})^T(s^o\mathbf{R})^{-1}(s^o\mathbf{R}\mathbf{D}^{-1})\mathbf{d} = s^o\Big(p - \mathrm{Tr}[\mathbf{H}\mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}]\Big) \tag{68}$$

$$s^o\mathbf{d}^T\mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}\mathbf{d} = s^o \mathrm{Tr}[\mathbf{R}(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}] \tag{69}$$

$$\mathbf{d}^T\mathbf{D}^{-1}\mathbf{R}\mathbf{D}^{-1}\mathbf{d} = \mathrm{Tr}[\mathbf{R}\mathbf{D}^{-1}]. \tag{70}$$

Therefore, the DI01 method is equivalent to a Maximum Likelihood estimation of $\mathbf{B}$ and $\mathbf{R}$ parameterized by s^b, s^o.

Bibliography

[1] NICC wildland fire summary and statistics annual report, 2021.

[2] A. Alexandridis, D. Vakalis, C. Siettos, and G. Bafas. 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*, 204(1):191–201, 2008.

[3] S. Al_Janabi, I. Al_Shourbaji, and M. A. Salman. Assessing the suitability of soft computing approaches for forest fires prediction. *Applied computing and informatics*, 14(2):214–224, 2018.

[4] M. Amendola, R. Arcucci, L. Mottet, C. Q. Casas, S. Fan, C. Pain, P. Linden, and Y.-K. Guo. Data assimilation in the latent space of a neural network, 2020.

[5] P. L. Andrews. Current status and future needs of the behaveplus fire modeling system. *International Journal of Wildland Fire*, 23(1):21–33, 2013.

[6] R. Arcucci, L. Mottet, C. Pain, and Y.-K. Guo. Optimal reduced space for variational data assimilation. *Journal of Computational Physics*, 379, 11 2018.

[7] R. Arcucci, J. Zhu, S. Hu, and Y.-K. Guo. Deep data assimilation: integrating deep learning with data assimilation. *Applied Sciences*, 11(3):1114, 2021.

[8] M. Asch, M. Bocquet, and M. Nodet. *Data assimilation: methods, algorithms, and applications*. Fundamentals of Algorithms. SIAM, 2016.

[9] S. Bao, N. Xiao, Z. Lai, H. Zhang, and C. Kim. Optimizing watchtower locations for forest fire monitoring using location models. *Fire safety journal*, 71:100–109, 2015.

[10] P. Barmpoutis, K. Dimitropoulos, K. Kaza, and N. Grammalidis. Fire detection from images using faster r-cnn and multidimensional texture analysis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8301–8305. IEEE, 2019.

[11] P. Becker, H. Pandya, G. Gebhardt, C. Zhao, C. J. Taylor, and G. Neumann. Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, pages 544–552. PMLR, 2019.

[12] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[13] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[14] R. E. Burgan. *Concepts and interpreted examples in advanced fuel modeling*, volume 238. US Department of Agriculture, Forest Service, Intermountain Research Station, 1987.

[15] L. Cai, H. S. He, Z. Wu, B. L. Lewis, and Y. Liang. Development of standard fuel models in boreal forests of northeast china through calibration and validation. *PLoS One*, 9(4):e94043, 2014.

[16] C. Q. Casas, R. Arcucci, P. Wu, C. Pain, and Y.-K. Guo. A reduced order deep data assimilation model. *Physica D: Nonlinear Phenomena*, 412:132615, 2020.

[17] B. Chapnik, G. Desroziers, F. Rabier, and O. Talagrand. Property and first application of an error-statistics tuning method in variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 130(601):2253 – 2275, 2004.

[18] B. Chapnik, G. Desroziers, F. Rabier, and O. Talagrand. Diagnosis and tuning of observational error in a quasi-operational data assimilation setting. *Quarterly Journal of the Royal Meteorological Society*, 132(615):543–565, 2006.

[19] M. Cheng, F. Fang, I. Navon, and C. Pain. A real-time flow forecasting with deep convolutional generative adversarial network: Application to flooding event in denmark. *Physics of Fluids*, 33(5):056602, 2021.

[20] S. Cheng, J.-P. Argaud, B. Iooss, D. Lucor, and A. Ponçot. Background error covariance iterative updating with invariant observation measures for data assimilation. *Stochastic Environmental Research and Risk Assessment*, 33(11):2033–2051, 2019.

[21] S. Cheng, J.-P. Argaud, B. Iooss, D. Lucor, and A. Ponçot. Error covariance tuning in variational data assimilation: application to an operating hydrological model. *Stochastic Environmental Research and Risk Assessment*, 35(5):1019–1038, 2021.

[22] S. Cheng, D. Lucor, and J.-P. Argaud. Observation data compression for variational assimilation of dynamical systems. *Journal of Computational Science*, page 101405, 2021.

[23] S. Cheng and M. Qiu. Observation error covariance specification in dynamical systems for data assimilation using recurrent neural networks. *Neural Computing and Applications*, pages 1–19, 2021.

[24] A. Cioaca and A. Sandu. Low-rank approximations for computing observation impact in 4D-Var data assimilation. *Computers & Mathematics with Applications*, 67(12):2112 – 2126, 2014.

[25] S. C. Coogan, L. D. Daniels, D. Boychuk, P. J. Burton, M. D. Flannigan, S. Gauthier, V. Kafka, J. S. Park, and B. M. Wotton. Fifty years of wildland fire science in canada. *Canadian Journal of Forest Research*, 51(2):283–302, 2021.

[26] G. Descombes, T. Auligné, F. Vandenberghe, D. Barker, and J. Barré. Generalized background error covariance matrix model (gen_be v2. 0). *Geoscientific Model Development*, 8(3):669–696, 2015.

[27] G. Desroziers, L. Berre, B. Chapnik, and P. Poli. Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3385 – 3396, 2005.

[28] G. Desroziers and S. Ivanov. Diagnosis and adaptive tuning of observation-error parameters in a variational assimilation. *Quarterly Journal of the Royal Meteorological Society*, 127(574):1433 – 1452, 2001.

[29] S. H. Doerr and C. Santín. Global trends in wildfire and its impacts: perceptions versus realities in a changing world. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1696):20150345, 2016.

[30] R. Dutta, A. Das, and J. Aryal. Big data integration shows australian bush-fire frequency is increasing significantly. *Royal Society open science*, 3(2):150241, 2016.

[31] R. Fablet, S. Ouala, and C. Herzet. Bilinear residual neural network for the identification and forecasting of dynamical systems. *arXiv preprint-dummy arXiv:1712.07003*, 2017.

[32] M. A. Finney. *FARSITE, Fire Area Simulator–model development and evaluation*. Number 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

[33] M. Fisher. Background error covariance modelling. In *Seminar on Recent developments in data assimilation for atmosphere and ocean (Shinfield Park, Reading, 8-12 September)*. ECMWF, 2003.

[34] W. Fulton. Eigenvalues, invariant factors, highest weights, and schubert calculus. *Bulletin of The American Mathematical Society*, 37:209–250, 2000.

[35] S. Ganapathi Subramanian and M. Crowley. Using spatial reinforcement learning to build forest wildfire dynamics models from satellite images. *Frontiers in ICT*, 5:6, 2018.

[36] M. G. Genton. Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, 2(Dec):299–312, 2001.

[37] L. Giglio, W. Schroeder, and C. O. Justice. The collection 6 modis active fire detection algorithm and fire products. *Remote Sensing of Environment*, 178:31–41, 2016.

[38] H. Gong, S. Cheng, Z. Chen, and L. Qing. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. *Nuclear Science and Engineering*, in press, 2022.

[39] E. Guelpa, A. Sciacovelli, V. Verda, and D. Ascoli. Model reduction approach for wildfire multi-scenario analysis. *Parte: http://hdl. handle. net/10316.2/34013*, 2014.

[40] E. Guelpa, A. Sciacovelli, V. Verda, and D. Ascoli. Faster prediction of wildfire behaviour by physical models through application of proper orthogonal decomposition. *International Journal of Wildland Fire*, 25(11):1181–1192, 2016.

[41] J. E. Hilton, J. E. Leonard, R. Blanchi, G. J. Newnham, K. Opie, A. Power, C. Rucinski, and W. Swedosh. Radiant heat flux modelling for wildfires. *Mathematics and Computers in Simulation*, 175:62–80, 2020.

[42] J. E. Hilton, A. L. Sullivan, W. Swedosh, J. Sharples, and C. Thomas. Incorporating convective feedback in wildfire simulations using pyrogenic potential. *Environmental Modelling & Software*, 107:12–24, 2018.

[43] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[44] X. Hu, Y. Sun, and L. Ntaimo. Devs-fire: design and application of formal discrete event wildfire spread and suppression models. *Simulation*, 88(3):259–279, 2012.

[45] X. Huang and G. Rein. Smouldering combustion of peat in wildfires: Inverse modelling of the drying and the thermal and oxidative decomposition kinetics. *Combustion and Flame*, 161(6):1633–1644, 2014.

[46] F. Huot, R. L. Hu, M. Ihme, Q. Wang, J. Burge, T. Lu, J. Hickey, Y.-F. Chen, and J. Anderson. Deep learning models for predicting wildfires from historical remote-sensing data. *arXiv preprint arXiv:2010.07445*, 2020.

[47] A. Jaafari, E. K. Zenner, M. Panahi, and H. Shahabi. Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheuristic optimization algorithms for spatial prediction of wildfire probability. *Agricultural and forest meteorology*, 266:198–207, 2019.

[48] W. Jahn, G. Rein, and J. L. Torero. Forecasting fire dynamics using inverse computational fluid dynamics and tangent linearisation. *Advances in Engineering Software*, 47(1):114–126, 2012.

[49] P. Jain, S. C. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, 2020.

[50] M. G. Just, M. G. Hohmann, and W. A. Hoffmann. Where fire stops: vegetation structure and microclimate influence fire spread along an ecotonal gradient. *Plant Ecology*, 217(6):631–644, 2016.

[51] H. Liang, M. Zhang, and H. Wang. A neural network model for wildfire scale prediction using meteorological factors. *IEEE Access*, 7:1–1, 12 2019.

[52] J. Mandel, L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek. A wildland fire model with data assimilation. *Mathematics and Computers in Simulation*, 79(3):584–606, 2008.

[53] N. Markuzon and S. Kolitz. Data driven approach to estimating fire danger from satellite images and weather information. In *2009 ieee applied imagery pattern recognition workshop*, pages 1–7. IEEE, 2009.

[54] R. Ménard. Error covariance estimation methods based on analysis residuals: theoretical foundation and convergence properties derived from simplified observation networks. *Quarterly Journal of the Royal Meteorological Society*, 142(694):257–273, 2016.

[55] G. Owen, J. D. McLeod, C. A. Kolden, D. B. Ferguson, and T. J. Brown. Wildfire management and forecasting fire potential: the roles of climate information and social networks in the southwest united states. *Weather, Climate, and Society*, 4(2):90–102, 2012.

[56] G. D. Papadopoulos and F.-N. Pavlidou. A comparative review on wildfire simulators. *IEEE systems Journal*, 5(2):233–243, 2011.

[57] D. F. Parrish and J. C. Derber. The National Meteorological Center's spectral statistical-interpolation analysis system. *Monthly Weather Review*, 120(8):1747–1763, 1992.

[58] E. Pastor, L. Zárate, E. Planas, and J. Arnaldos. Mathematical models and calculation systems for the study of wildland fire behaviour. *Progress in Energy and Combustion Science*, 29(2):139–153, 2003.

[59] S. Pawar, S. E. Ahmed, O. San, A. Rasheed, and I. M. Navon. Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows. *Physics of Fluids*, 32(7):076606, 2020.

[60] G. Perry. Current approaches to modelling the spread of wildland fire: a review. *Progress in Physical Geography*, 22(2):222–245, 1998.

[61] C. Quilodrán-Casas, R. Arcucci, C. Pain, and Y. Guo. Adversarially trained lstms on reduced order models of urban air pollution simulations. *arXiv preprint arXiv:2101.01568*, 2021.

[62] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[63] O. Rios, E. Pastor, M. Valero, and E. Planas. Short-term fire front spread prediction using inverse modelling and airborne infrared images. *International Journal of Wildland Fire*, 25(10):1033–1047, 2016.

[64] O. Rios, M. M. Valero, E. Pastor, and E. Planas. A data-driven fire spread simulator: validation in vall-llobrega's fire. *Frontiers in Mechanical Engineering*, 5:8, 2019.

[65] M. C. Rochoux, S. Ricci, D. Lucor, B. Cuenot, and A. Trouvé. Towards predictive data-driven simulations of wildfire spread–part i: Reduced-cost ensemble kalman filter based on a polynomial chaos surrogate model for parameter estimation. *Natural Hazards and Earth System Sciences*, 14(11):2951–2973, 2014.

[66] D. Rodriguez-Aseretto, D. De Rigo, M. Di Leo, A. Cortés, and J. San-Miguel-Ayanz. A data-driven model for large wildfire behaviour prediction in europe. *Procedia Computer Science*, 18:1861–1870, 2013.

[67] R. C. Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest & Range Experiment Station, Forest Service, US, 1972.

[68] O. San, R. Maulik, and M. Ahmed. An artificial neural network framework for reduced order modeling of transient flows. *Communications in Nonlinear Science and Numerical Simulation*, 77:271–287, 2019.

[69] E. Scaduto, B. Chen, and Y. Jin. 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*, 13:5102–5114, 2020.

[70] W. Schroeder, P. Oliva, L. Giglio, and I. A. Csiszar. The new VIIRS 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, 143:85–96, 2014.

[71] X. Silvani, F. Morandini, and J.-L. Dupuy. Effects of slope on fire spread observed through video images and multiple-point thermal measurements. *Experimental Thermal and Fluid Science*, 41:99–111, 2012.

[72] L. Sirovich. Turbulence and the dynamics of coherent structures. i. coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[73] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[74] O. Talagrand. A posteriori evaluation and verification of analysis and assimilation algorithms. In *Workshop on Diagnosis of Data Assimilation Systems*, pages 17–28, Shinfield Park, Reading, 1998.

[75] P. Tandeo, P. Ailliot, M. Bocquet, A. Carrassi, T. Miyoshi, M. Pulido, and Y. Zhen. A review of innovation-based methods to jointly estimate model and observation error covariance matrices in ensemble data assimilation. *Monthly Weather Review*, 148(10):3973–3994, 2020.

[76] D. Wang, D. Guan, S. Zhu, M. Mac Kinnon, G. Geng, Q. Zhang, H. Zheng, T. Lei, S. Shao, P. Gong, et al. Economic footprint of california wildfires in 2018. *Nature Sustainability*, 4(3):252–260, 2021.

[77] D. Weise, J. Cobian-Iñiguez, and M. Princevac. Surface to crown transition. In *S. L. Manzello, ed. Encyclopedia of Wildfires and Wildland-Urban Interface (WUI) Fires*, Cham, Switzerland, 2018. Springer, Cham. 5 p.

[78] D. R. Weise and G. S. Biging. A qualitative comparison of fire spread models incorporating wind and slope effects. *Forest Science*, 43(2):170–180, 1997.

[79] R. E. Wolfe, G. Lin, M. Nishihama, K. P. Tewari, J. C. Tilton, and A. R. Isaacman. Suomi npp viirs prelaunch and on-orbit geometric calibration and characterization. *Journal of Geophysical Research: Atmospheres*, 118(20):11–508, 2013.

[80] S. D. Wong, J. C. Broader, and S. A. Shaheen. Review of california wildfire evacuations from 2017 to 2019. Technical report, University of California, 2020.

[81] W. Yu, J. Gonzalez, and X. Li. Fast training of deep lstm networks with guaranteed stability for nonlinear system modeling. *Neurocomputing*, 422:85–94, 2021.

[82] G. Zhang, M. Wang, and K. Liu. Forest fire susceptibility modeling using a convolutional neural network for yunnan province of china. *International Journal of Disaster Risk Science*, 10(3):386–403, 2019.

[83] Q.-x. Zhang, G.-h. Lin, Y.-m. Zhang, G. Xu, and J.-j. Wang. Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images. *Procedia engineering*, 211:441–446, 2018.

[84] Q. Zhu, F. Li, W. J. Riley, L. Xu, L. Zhao, K. Yuan, H. Wu, J. Gong, and J. T. Randerson. Building a machine learning surrogate model for wildfire activities within a global earth system model. *Geoscientific Model Development Discussions*, pages 1–22, 2021.