Imperial College London
Department of Earth Science and Engineering
MSc in Applying Computational Science and Engineering

Independent Research Project
Final Report

Reduced order modelling and latent data assimilation for wildfire probability prediction

by

Caili Zhong

Email: caili.zhong21@imperial.ac.uk
GitHub username: acse-cz421
Repository: https://github.com/ese-msc-2021/irp-cz421
Supervisors:
Dr. Sibo Cheng
Dr. Matthew Kasoar

July/September 2022

Abstract

The occurrence of forest fires can cause severe damage to the vegetation as well as to the soil in the ecosystem and can also indirectly affect the climate. The analysis of the interaction between wildfires and environmental factors is used to predict the occurrence of fires, which is the current JULES-INFERNO project. However, this prediction project uses mathematical methods to simulate vegetation and the environment, so that it can suffer from modelling difficulties and high computational costs. Deep learning models have an advantage in handling large amounts of data. They can reduce the computational cost of subsequent prediction models by extracting data features through Reduced Order Modelling (ROM) and compressing the data to a low-dimensional latent space. Therefore, this study proposes a JULES-INFERNO-based surrogate model based on ROM and deep learning prediction networks to improve the efficiency of physical wildfire prediction models. In addition, the model implements iterative prediction. In order to avoid the accumulation of errors from iterative prediction, Latent data Assimilation (LA) is applied to the prediction process to periodically adjust the prediction results to ensure stability and sustainability of the prediction. The results show that the surrogate model can effectively encode the original data and achieve wildfire prediction in a short period. Furthermore, the application of LA can also effectively adjust the bias of the prediction results.

Acknowledgements

I have met many people to thank during these three months research.

Firstly, I would like to thank my supervisor, Dr. Sibo Cheng, for his patient teaching and impressive academic achievements. Thank you for his selflessness in sharing his technical and academic skills and knowledge with me. He tireless inquiries about the progress of my projects and his patience in guiding the direction of my work. His easy-going style is also worth learning from. Finally, thank him for giving me a goal to strive for.

Secondly, I would like to thank Dr. Marrhew Kasoar, who was not my first supervisor but provided the dataset for this project, advised me on improvements in this research and the thesis. I would like to express my sincere gratitude and respect to him.

Finally, I would like to thank my classmates Hansong Xiao, Yanan Xiong, Tengteng Huang, Lu Li, Shiqi Yin, Haotong Jin, Haoran Cheng, Yanqi Huo and Zheng Xing for our companionship and communication that allowed us to progress together. I would also like to express my sincere gratitude to all supervisors reviewing this paper.

Contents

List of Tables

List of Figures

List of Acronyms

DGVM	Dynamic Global Vegetation Model
JULES-INFERNO	Joint UK Land Environment Simulator - INteractive Fire and Emissions algorithm for Natural envirOnments
HPC	High-performance Computing
ROM	Reduced Order Modelling
Seq2Seq	Sequence-to-Sequence
DA	Data Assimilation
LA	Latent Data Assimilation
CAE	Convolutional Autoencoder
PCA	Principal Component Analysis
LSTM	Long Short Term Memory
AE	Autoencoder
RNN	Recurrent Neural Network

Chapter 1
Introduction

1.1 Motivation

Fire is one of the most destructive disasters, and every year fires have significant economic, social, and environmental impacts on a global scale, especially in the case of forest fires, which are often sudden and extraordinarily destructive and difficult to deal with immediately [1]. Globally, an average of more than 200,000 forest fires occurs each year, burning more than 1% of the world's forested area [2]. Wildfires not only destroy trees but also cause severe damage to the forest environment and ecological structure. Furthermore, the physicochemical properties of the soil will also change with the occurrence of wildfire, which can result in serious ecosystem imbalances [3]. Therefore, it is essential to study the influence of vegetation and climatic factors on the occurrence of wildfires, which can anticipate future trends through environmental factors and then analyze the interactions between wildfires and vegetation to control the ecological balance. The main objective of this study is to predict the global wildfire occurrence probability.

Furthermore, global fire frequency is related to land use, vegetation type and meteorological factors. For instance, when fire risk weather meets combustible vegetation, it may cause wildfires, and fires can damage vegetation and soil health. Thus, data from validated natural environment models can be used to predict about forest fires. Numerous valid and relevant models have been constructed, such as Earth System Models [4] and Dynamic Global Vegetation Models (DGVM) [5]. The Joint UK Land Environment Simulator - INteractive Fire and Emissions algorithm for Natural envirOnments (JULES-INFERNO) is an example of the DGVM, which can effectively simulate the probability of fire occurrence based on geographic features (vegetation, climate, etc.) [6]. However, implementing such a surface simulation is highly challenging due to the complexity of the physical model and geographical features. For example, predicting the probability wildfires over a 100-year climate change scenario would take approximate a week with 32 threads on the Jasmin national High-performance Computing (HPC) resource [7]. Therefore, it is necessary to build an efficient surrogate model to address this issue.

Since the 1990s, artificial intelligence has been applied to the study of vegetation and wildfire, in addition to traditional statistical and physical models [8]. Much research has recently used Reduced Order Modelling (ROM) to reduce the computational stress of surrogate model. That is because that ROM could compress the raw data into a low-dimensional space with an encoder and then restore the data using a decoder. [9]. Thus, predictions and simulations can be performed in a low-dimensional latent space and then decoded afterwards. However, prediction models trained using large amounts of data do not necessarily guarantee accuracy in long-term prediction. For instance, using a Sequence-to-Sequence (Seq2seq) forecasting model to make predictions based on existing data, the error accumulates over time if the predictions are not corrected as the iterations progress. In order to address this challenge, many industries have applied data assimilation (DA) methods, which combine simulated data with observed actual data through specific weighting, and then correct the predicted data [10]. Additionally, DA has been widely applied in high-dimensional dynamical systems, such as weather forecasting and climate simulation [11]. However, with the increasing number of meteorological satellites, the system model can suffer from the extremely high data dimension, especially for the multi-dimension system [11]. Directly applying the full-size data with DA operations could lead to high computational costs, so experts have proposed Latent data Assimilation

(LA) [12]. LA combines ROM and DA; the data is compressed before the DA operation is performed [12]. Therefore, the algorithm scheme proposed in this study will combine ROM, machine learning predictive models, and LA.

1.2 Aims

The main objective of this study is to propose a deep learning surrogate model based on JULES-INFERNO with the same input and output requirements as JULES-INFERNO. After the model is built and trained, a time series of wildfire-related data is input to predict the occurrence of wildfires in future years. This surrogate model can significantly improve prediction efficiency compared to the physical model. This model combines the JULES-INFERNO model with a deep neural network. Data from the JULES-INFERNO simulations is used as input to train the machine learning model. Testing the model's effect with unknown scenarios for the ultimate purpose of making predictions about future fire conditions efficiently.

However, the data of the JULES-INFERNO project is on a global scale, and multiple variables were used in this study. Thus, the data feature size is significant, and the computational cost would be high if it is directly used as the input of the prediction model. Reducing the dimensionality of the data in advance could address this issue, i.e., ROM. Therefore, two ROM approaches, Convolutional Autoencoder (CAE) and Principal Component Analysis (PCA), were chosen for this research to reduce the data dimensionality. In terms of the wildfire occurrence probability prediction, this research aims to implement a machine learning prediction model with Long Short Term Memory (LSTM) as the main structure, as LSTM is suitable for dealing with problems highly correlated with time series [13]. In addition, for the prediction model to have sustainable prediction capability, LA was applied to correct the model results, and the observation data of LA is the actual encoded data. Compared to the traditional DA in the entire physical space, LA can considerably improve computational efficiency thanks to ROM [13]. Overall, this research aims to implement a global-scale forest wildfire prediction model that combines ROM, recurrent neural networks (LSTM) and LA for effective wildfire forecasting, which uses JULES-INFERNO data as input.

1.3 Main Contribution

This research makes the following contributions:

- This research implements a deep learning surrogate model for wildfire prediction based on JULES-INFERNO, which can effectively improve the prediction efficiency of the physical model with the same input and output requirements. The physical model consumes significant time and computational costs in forecasting due to the complexity of the physical model and the variability of the actual terrain and weather. However, a machine learning surrogate model can save computational costs by embedding the input into a low-dimensional space before prediction. Specifically, the model's implementation includes ROM, predictive modelling, and LA.

- Regarding ROM, CAE was constructed in this study for the climate and wildfire data, which allows for a significant reduction in data dimensionality to improve the efficiency of subsequent predictions while maintaining the original characteristics of the data. Furthermore, this research also implements a PCA model for order reduction. When both methods were tested using the unseen dataset, it was found that the errors in the CAE reconstructed data were smaller than in the PCA.

- The mechanism of the surrogate model is making iterative predictions, which use the current predicted results as next-level input to achieve a long-term prediction. In order to avoid error accumulation, this research has applied LA, which used the original encoded data as observation, and weighted the predicted values with the observed values to correct the predicted values periodically. When training a surrogate model with a test set (unseen scenarios), using LA to adjust the predicted values can effectively improve the accuracy of subsequent predictions compared to not using the LA method. Therefore, the LA implemented in this project can effectively stabilize prediction results.

The paper is organized as follows: explaining the original physical model, JULES-INFERNO, and the data set of this project in Section 2. In Section 3, the dimensionality reduction method used in the study (PCA and the constructed CAE structure) is described, followed by an explanation of the surrogate model used in this research for wildfire prediction, and finally, a description of the principles and applications of the LA in this research. Section 4 discusses the experimental results and analysis of the study. Finally, this research concludes and suggests future directions for enhancement in section 5.

Chapter 2
JULES-INFERNO

JULES is a model that can simulate land surface vegetation, primarily using physical models to simulate the processes of land-use, fire, and climate interactions with vegetation dynamics. However, the model formulation of disturbances, particularly fire, drought, and tree mortality, is not sufficiently constrained [14]. The INFERNO fire scheme was constructed based on a simplified parameterization of fire counts. Additionally, it can efficiently predict large-scale fires based on climatology variables [15]. In 2016, JULES invoked a new fire disturbance term from the INFERNO model, and the new JULES-INFERNO model can effectively represent the complexity and interaction between land-use change and fire [6]. Precisely, the model will calculate flammability based on natural conditions such as soil, vegetation that can be fuel, temperature, humidity, and precipitation, and then combine lightning as well as population density to simulate the average area burnt. Finally, emitted atmospheric aerosols and trace gases can be calculated from the average burnt area and vegetation-dependent emission factors [6]. Figure 2.1 illustrates this project's inputs and critical components [16]. However, such a model involves a large amount of data and parameters, so there is a high calculation pressure during the simulation, which could make the prediction inefficient. A method to improve this situation is to use a surrogate model that substitutes the original high-precision simulation model. The surrogate model aims to use the exact input requirements as the original model and output a result that approximates the original model but is less computationally intensive to solve [17].

This research applied four meteorological boundary conditions (temperature (T), humidity (H), rainfall (R), and lightning (L)) and an experimental variable (Grid box mean burnt area fraction ('burnt_area_gb')) from the JULES-INFERNO as input data. In terms of each meteorological boundary condition, the observations are from 1961 to 1990 and each year was collected monthly, so there are 360 snapshots for each variable. Then, the snapshots in each parameter can be denoted as temperature ($T_i, i = 1, 2, \cdots, 360$), humidity ($H_i, i = 1, 2, \cdots, 360$), rainfall ($R_i, i = 1, 2, \cdots, 360$), and lightning ($L_i, i = 1, 2, \cdots, 360$). Additionally, each snapshot observation has a latitude span of 144 units and a longitude span of 192 units, so the size of each one is 144×192, for instance, the shape of T is (360, 144, 192). Regarding 'burnt_area_gb', this project used five sets of experimental results ($P_s, s = 1, 2, \cdots, 5$), and each set was driven by the same climatic conditions (1961 to 1990). However, JULES-INFERNO applied different initial internal states to ensure the robustness of its model; the internal state of P_1 was randomized, and the initial internal states of the subsequent experimental results were all the last internal states of the previous one. Therefore, although each result was for 1961-1990, there was variability, and for each data can be denoted as $P_{s,i}, s = 1, 2, \cdots, 5, i = 1, 2, \cdots, 360$. Different from the climatic conditions, the fire data only spans 112 latitude units, so the fire snapshot size is 112×192. Additionally, all data used in this study are from the JULES-INFERNO project [16].

In subsequent experiments, P_1, P_2 and portion of P_3 are the training set and another portion of P_3 is the validation set during training. P_4 and P_5 were used as the test set for all models built in the study. In addition, to eliminate the adverse effects of odd sample data in training, this research standardized all training and test sets (climates and fire variables). Apply Equation (1) to normalize the data to [0, 1]:

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

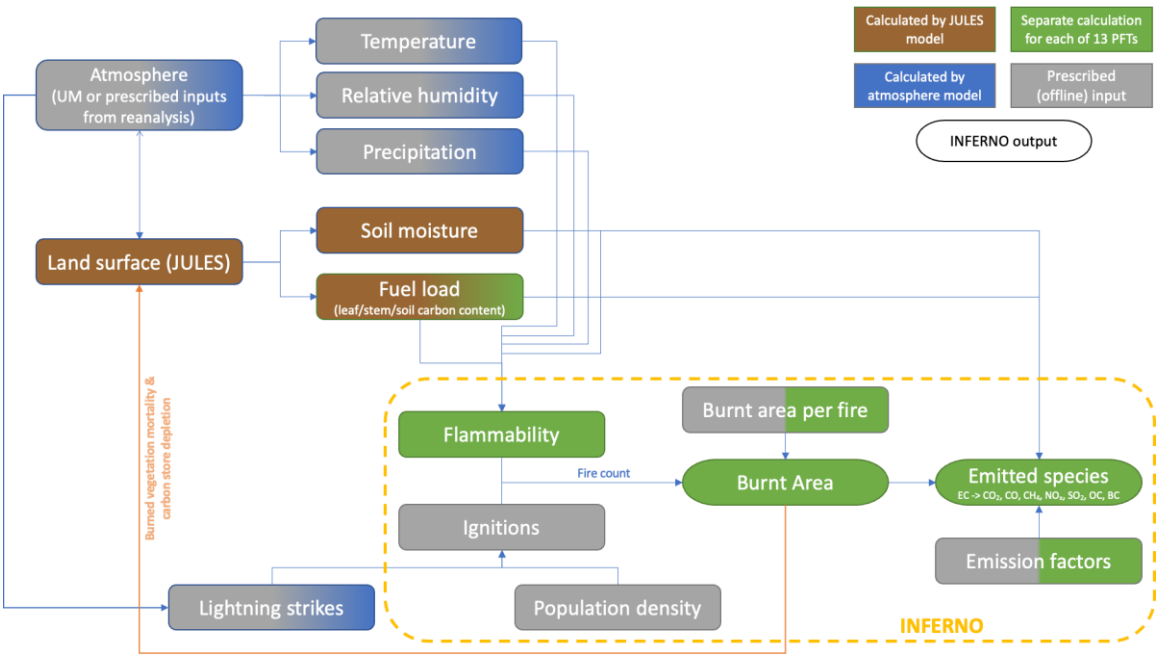


Figure 2.1 The mechanism of JULES-INFERNO [16]

Chapter 3
Methodology

The research started by constructing the ROM to reduce the dimension of JULES-INFERNO data. The climate and wildfire data are then stitched together and processed into time-series data fed into the prediction model for training. Then, testing the trained surrogate model by iterative forecasting, which uses one year of unseen data as input to predict fires for the next 29 years. Finally, LA is used periodically during the forecasting process to adjust the forecasts.

3.1 Reduced order modelling

When extracting data features, if all features are selected for trend prediction it may not only increase the complexity of the calculation significantly, but also leads to a large amount of redundancy in the data [18]. Therefore, this research built PCA and CAE models for dimension reduction.

Assume that the original data is an n-dimensional vector noted as $U = \{u_i\}_{i=1,2,\cdots,n}$. The compressed variable (k-dimension, $k < n$) after dimensionality reduction is denoted as $\tilde{U} = \{\tilde{u}_i\}_{i=1,2,\cdots,k}$. The variable that reconstructs the compressed data to its original size is denoted as $\hat{U} = \{\hat{u}_i\}_{i=1,2,\cdots,n}$. ROM's primary purpose is to minimize expectation of the Mean Square Error (MSE) $E\left[\left(U - \hat{U}\right)^2\right]$. Climate data and fire data obtained by dimensionality reduction could be denoted by $\widetilde{T_i}, \widetilde{H_i}, \widetilde{R_i}, \widetilde{L_i}, \widetilde{P_{s,i}}, s = 1,2,\cdots 5;\ i = 1,2,\cdots 360$, and each snapshot is k-dimension. Decoded data is denoted as $\widehat{T_i}, \widehat{H_i}, \widehat{R_i}, \widehat{L_i}, \widehat{P_{s,i}}, s = 1,2,\cdots 5;\ i = 1,2,\cdots 360$, and dimensionality of each element consistent with the original data (n-dimension).

3.1.1 PCA

PCA uses the idea of dimensionality reduction to form new variables by linearly combining multiple parameter indicators, mapping the original n-dimensional features to k-dimensional features ($n > k$), k is known as the truncation parameter in PCA [18]. The principles of PCA are shown in Appendix A.

3.1.2 CAE

An Autoencoder (AE) is a self-supervised algorithm which minimizes the reconstruction error between its model input and output by optimizing parameters to obtain a low-dimensional data representation of high-dimensional data features [19]. The feature extraction method used in standard AE is fully connected. Each neuron in each layer is connected to all the neurons in the next layer. Thus, it generates massive parameters, making the computation more expensive and ignoring some of the features in the image [20]. To address the drawbacks of standard AE, experts have proposed the CAE [20], a combination of AE and CNN. The CAE model inherits the self-supervision function of standard AE, replacing the matrix product operation between hidden neurons with convolution and pooling operations for capturing local spatial patterns of monitoring data [20]. Furthermore, extracting features by convolution can reduce the use of parameters and increase the training speed.

The climate and fire conditions vary from region to region in this study, and the data is a two-dimensional matrix (which can be represented as an image). Therefore, this research constructed a CAE model to achieve self-supervised learning of environmental and wildfire features. In addition, the reconstruction loss is used to evaluate the model performance.

The encoder designed for this study uses three convolutional layers, three max-pooling layers and two fully connected layers. The decoder includes four convolutional layers, three up-sampling layers and one fully connected layer. Figure 3.1 shows the CAE structure of this study with a latent space of dimension 100.

Firstly, the sample data, $U = \{u_i\}_{i=1,2,\cdots,n}$ need to be converted into a two-dimensional matrix of p rows and q columns ($p * q = n$) according to the latitude and longitude range of the original data, and then each snapshot could be interpreted as an image. Given the input U, features are extracted by multiple convolutional kernels to obtain the output C_l of the l^{th} layer, then the $\mathrm{C}_l{}'$ is obtained by down-sampling through the subsequent pooling layer to retain the main features of the inflow data and prevent over-fitting. The operational methods are shown in Equations (2) and (3).

$$\begin{cases} \mathrm{C}_l = f(W_l \otimes U + b_l), & l = 1 \\ \mathrm{C}_l = f(W_l \otimes \mathrm{C}_{l-1}{}' + b_l), & l = 2,3 \end{cases} \tag{2}$$

$$\mathrm{C}_{l-1}{}' = Maxpooling(\mathrm{C}_l), \qquad l = 1,2,3 \tag{3}$$

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

As shown in Figure 3.1, during the network training process, the encoder acquires the feature mapping of the input data layer by layer. It obtains the k-dimensional latent space data after the Flatten and Dense layers. In contrast, the decoder inputs the latent space data and reconstructs the data by convolution, up sampling, etc., and evaluates the reconstruction performance of CAE on weather or wildfire data by the loss function (MSE). Equation (4) shows the square error of one sample ($L(\cdot)$) and mean error for training set ($J(\cdot)$), suppose the training set is P_{tr} with N_{tr} snapshots:

$$\begin{cases} L(U, \widehat{U}) = (U - \widehat{U})^2 \\ J(P_{tr}, \widehat{P_{tr}}) = \dfrac{1}{N_{tr}} \displaystyle\sum_{i=1}^{N_{tr}} L(P_{tr,i}, \widehat{P_{tr,i}}) \end{cases} \tag{4}$$

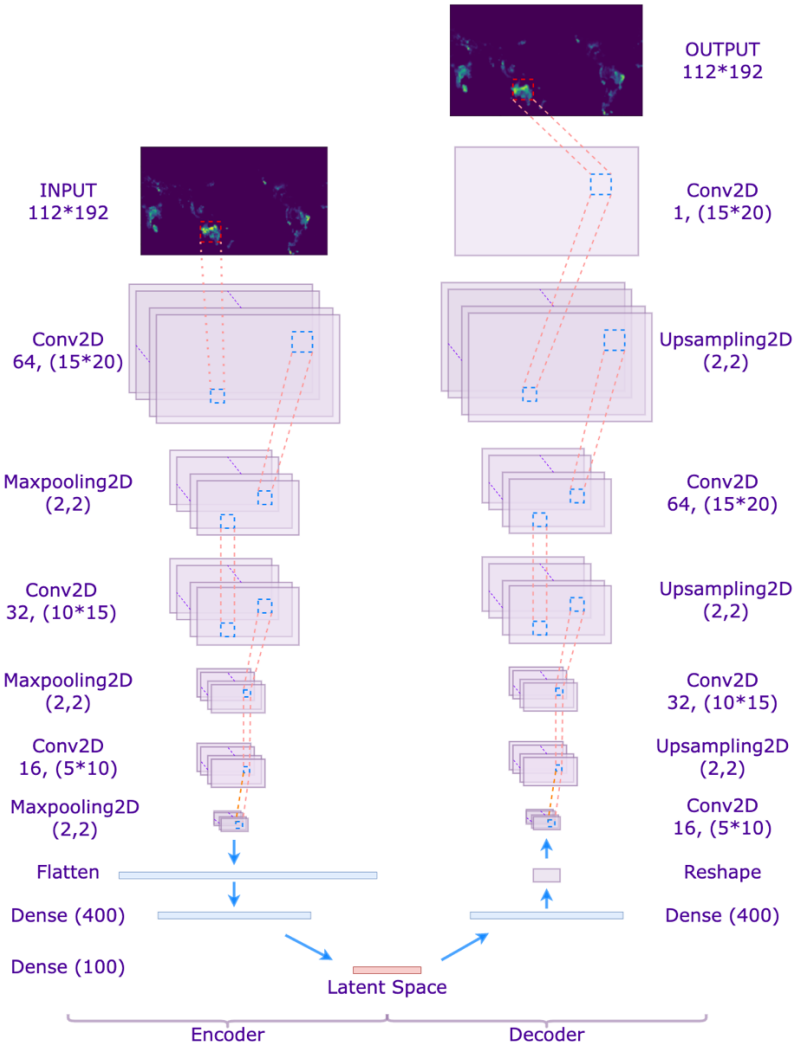


Figure 3.1 The CAE Model structure: Conv2d, MaxPooling2D, Dense, UpSampling2D representing the 2D convolutional layer, max pooling layer, fully connected layer, and up-sampling layer respectively

3.2 Surrogate predictive model

This research constructed 5 CAE models for different variables (climate and fire data) and obtained k-dimensional latent space for each of them respectively. Then, it combined the climate and fire data as inputs to the predictive surrogate model denoted as $Z = \{Z_s\}_{s=1,2,\cdots 5}$, and $Z_s = \{Z_{s,i}\}_{s=1,2,\cdots 5;\, i=1,2,\cdots 360}$, 's' indicates the 5 fire data sets respectively. In addition, each fire data is derived with the same climate data, so that for each sample could be denoted as $Z_{s,i} = [\widetilde{P_{s,i}}\ \widetilde{T_i}\ \widetilde{H_i}\ \widetilde{R_i}\ \widetilde{L_i}]$, $s = 1,2,\cdots 5; i = 1,2,\cdots 360$, with dimension of $4k$ for each snapshot. In addition, the data was normalised before it is fed into the model.

This study aims to build a Machine Learning model to surrogate the original physical prediction model. The Recurrent Neural Network (RNN) is an effective method for time series problems. However, it has long-term dependence problems for long time series, and gradient disappearance and explosion problems can

also occur during the backpropagation to update parameters [21]. The LSTM is a variant of the RNN, a special kind of neural network with three "gates" that could solve the problem of a long-term dependence on time series and is widely used in engineering applications [22]. In this research, LSTM network was used to build the wildfire prediction model with the JULES-INFERNO climate and fire data as inputs. Figure 3.2 shows the basic structure of the whole surrogate model.

The seq2seq prediction based on the LSTM network is implemented in this study with t_{in} timesteps as input and t_{out} timesteps as output. This research sets the $t_{in} = t_{out} = 12$, as the data is collected by month and 12 timesteps can represented a year. The training set of the LSTM network is $Z_{train} = Z_1 \cup Z_2 \cup Z_3$ and the test set is denoted as $Z_{test} = Z_4 \cup Z_5$. In order to enhance the dataset, the input to the LSTM training can be increased by shifting the initial time of each time series, as shown in Equation (5). The model uses MSE loss function to evaluate the training performance. The principles of LSTM are shown in Appendix A.

$$\begin{gathered}
\left[Z_{1,1}, Z_{1,2}, \cdots, Z_{1,t_{in}}\right] \xrightarrow{LSTM\ train} \left[Z_{1,t_{in}+1}, Z_{1,t_{in}+2}, \cdots, Z_{1,t_{in}+t_{out}}\right], \\
\left[Z_{1,2}, Z_{1,3}, \cdots, Z_{1,t_{in}+1}\right] \xrightarrow{LSTM\ train} \left[Z_{1,t_{in}+2}, Z_{1,t_{in}+3}, \cdots, Z_{1,t_{in}+t_{out}+1}\right] \\
\vdots \\
\left[Z_{1,360-t_{in}-t_{out}+1}, \cdots, Z_{1,360-t_{out}}\right] \xrightarrow{LSTM\ train} \left[Z_{1,360-t_{out}+1}, Z_{1,360-t_{out}+2}, \cdots, Z_{1,360}\right] \\
\vdots \\
\left[Z_{3,360-t_{in}-t_{out}+1}, \cdots, Z_{3,360-t_{out}}\right] \xrightarrow{LSTM\ train} \left[Z_{3,360-t_{out}+1}, Z_{3,360-t_{out}+2} \cdots, Z_{3,360}\right].
\end{gathered} \tag{5}$$

As for the iterative prediction in test data, the trained model uses the current predictive result as the input of next prediction. The predictive result can be denoted as the $Z'_{test} = Z'_4 \cup Z'_5$. The first-year data of the test set is used as model input, and the prediction result set only contains data from the following 29 years. Each prediction can predict the following year's wildfire, so the model only takes 29 timesteps to predict 29 years of data. The iterative prediction process is shown in Equation (6).

$$\begin{gathered}
\left[Z_{4,1}, Z_{4,2}, \cdots, Z_{4,12}\right] \xrightarrow{LSTM\ predict} \left[Z'_{4,13}, Z'_{4,14}, \cdots, Z'_{4,24}\right], \\
\left[Z'_{4,13}, Z'_{4,14}, \cdots, Z'_{4,24}\right] \xrightarrow{LSTM\ predict} \left[Z'_{4,25}, Z'_{4,26}, \cdots, Z'_{4,36}\right] \\
\vdots \\
\left[Z'_{4,337}, Z'_{4,338}, \cdots, Z'_{4,348}\right] \xrightarrow{LSTM\ predict} \left[Z'_{4,349}, Z'_{4,350}, \cdots, Z'_{4,360}\right], \\
\left[Z_{5,1}, Z_{5,2}, \cdots, Z_{5,12}\right] \xrightarrow{LSTM\ predict} \left[Z'_{5,13}, Z'_{5,14}, \cdots, Z'_{5,24}\right] \\
\vdots \\
\left[Z'_{5,337}, Z'_{5,338}, \cdots, Z'_{5,348}\right] \xrightarrow{LSTM\ predict} \left[Z'_{5,349}, Z'_{5,350}, \cdots, Z'_{5,360}\right].
\end{gathered} \tag{6}$$

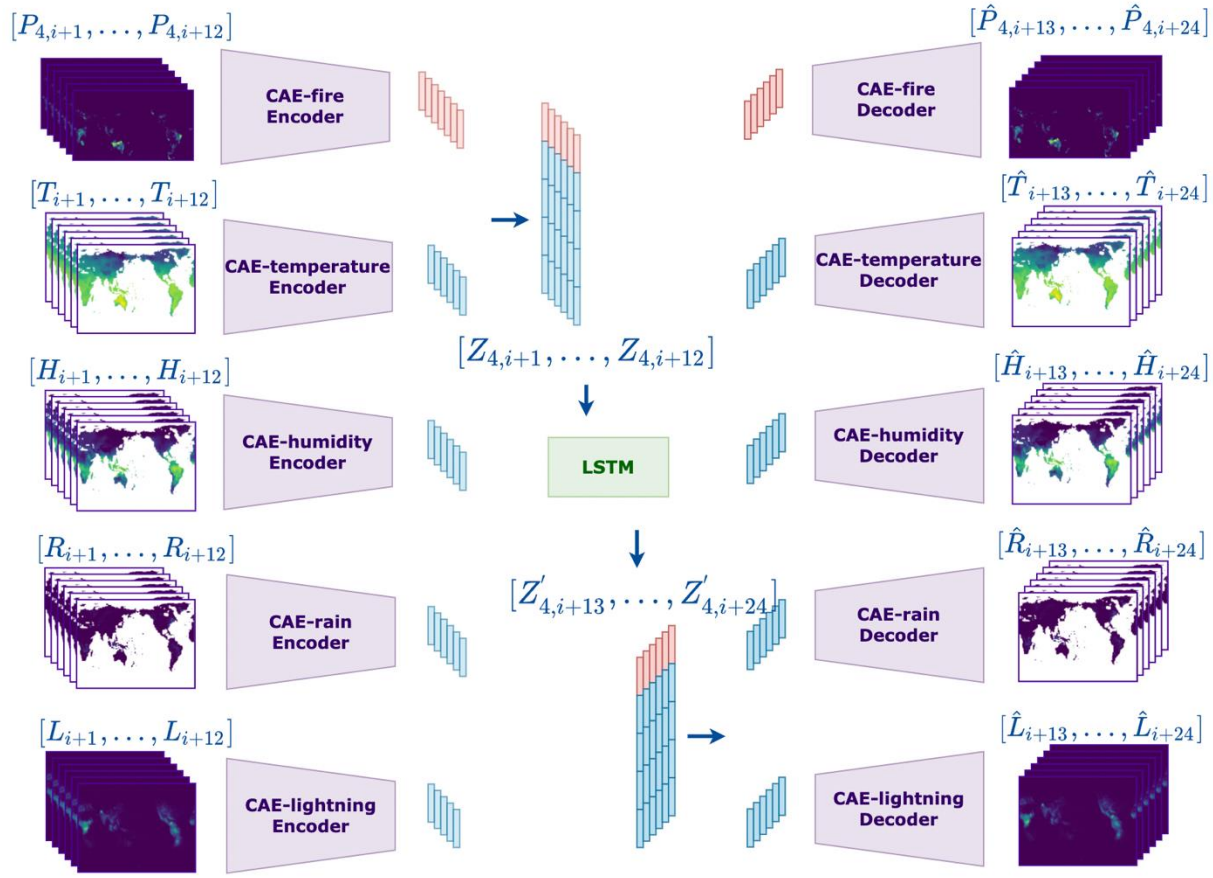


Figure 3.2 The Surrogate Model structure

3.3 Latent Data Assimilation

The basic principle of DA is the combination of numerical models with observational data to improve the simulation and forecasting of the system under study [23]. During the model prediction process, the study applies the DA method periodically to enhance the current prediction results with the help of the observation data. The 'observation' data in this research is considered the existing output of the JULES-INFERNO model. As periodic corrections are made to the forecast data, DA allows for a more stable and accurate long-term prediction. In this study, LA was applied, which combines ROM with DA, i.e., reducing the dimensionality of the data before applying the DA method [12]. In addition, ROM could reduce the parameters for subsequent operations, which allows LA has a significant advantage in terms of computational efficiency compared to the classical full-space DA.

Commonly used DA include variational approaches [24][25] and Kalman filtering methods [26]. The former one is widely used for data assimilation of atmospheric models and has a high computational efficiency [27]. Therefore, variational approach is used determine the assimilated latent variables in this study.

DA can be summarized as a problem of solving the minimization of an objective function $J_{DA}(\cdot)$ characterizing the deviation between the analysis field (optimal data $\dot{Z_{test}}$) and the observation field (actual data Z_{test}) as well as the background field (predicted data Z'_{test}), as shown in Equation (7).

$$J_{DA}(\dot{z_t}) = \frac{1}{2}(\dot{z_t} - z'_t)^T \mathrm{B}^{-1}(\dot{z_t} - z'_t) + \frac{1}{2}[z_t - \mathcal{H}(\dot{z_t})]^T \mathrm{O}^{-1}[z_t - \mathcal{H}(\dot{z_t})] \quad (7)$$

where t indicates the temporal index (the t^{th} predicted year), $\dot{z_t}$ is the analysis variable of DA, z'_t is the background variable, z_t is the observation (original) variable, B and O represent the background error covariance matrix and the observation error covariance matrix respectively. The modelling of this covariance determines the weight of prediction and observation in the objective function, which is crucial for DA approaches [28]. $\mathcal{H}(\cdot)$ is the state-observation function in latent space, which maps the analysis variable to the observation variable.

The control variable for the final optimization result could be obtained by optimizing the objective function (Equation (7)), which is the optimize analysis field $\dot{Z_{test,t}}$:

$$\dot{Z_{test,t}} = argmin\big(J_{DA}(\dot{z_t})\big) \quad (8)$$

this equation can be solved by the Best Linear Unbiased Estimator [29], since the transformation operator H could be approximated by linear function $\mathcal{H}(\cdot)$:

$$\dot{Z_{test,t}} = z'_t + \mathrm{BH}^T(\mathrm{HBH}^T + \mathrm{O})^{-1}\big(z_t - \mathcal{H}(z'_t)\big) \quad (9)$$

The obtained analysis state $\dot{Z_{test,t}}$ can be used as an initial point for the next level prediction, as shown in Figure 3.3, the output of LSTM could be $\dot{Z_{test,t}}$ (apply LA at t time index) or $Z'_{test,t}$ (LSTM prediction at t time index). These processes can be repeated periodically to consistently improve the prediction performance. Furthermore, in Figure 3.3, the final result of the surrogate model is the combination of the $\{\dot{Z_{test,n}}\}_{n\in[0,29]}$ and $\{Z'_{test,m}\}_{m\in[0,29]}$ $(m \neq n)$, which could be denoted as Z'_{opt}.

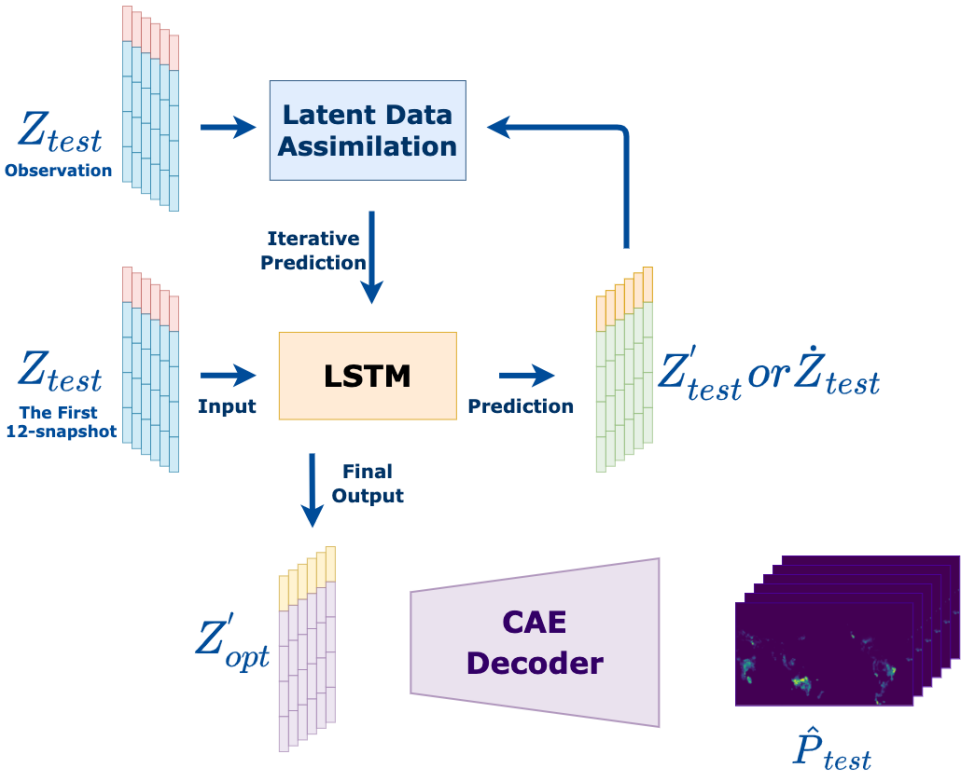


Figure 3.3 The Predict Model with LA

Chapter 4
Results

4.1 Code Metadata

For the development of this research, the hardware and software used are listed below:

- Programming language: Python (3.5 or higher)
- Platform: Google Colab Pro
- GPU: P100
- Installation requirements: os, pandas, math, numpy, netCDF4, seaborn, matplotlib, tensorflow, keras, sklearn

4.2 ROM results

In order to set a reasonable latent space dimensionality, the research examined the explained variance of PCA for this experimental dataset with different numbers of principal components, as shown in Figure 4.1. Generally, 85% to 95% of the explained variance can reflect most data information while effectively reducing the original data dimensionality [30]. Therefore, to reduce the latent space dimension as much as possible and the computational cost of subsequent processing, the research chose the 100-dimension with 87% explained variance, as shown in Figure 4.1. In addition, the 20-dimensional latent space is chosen for comparison purpose.

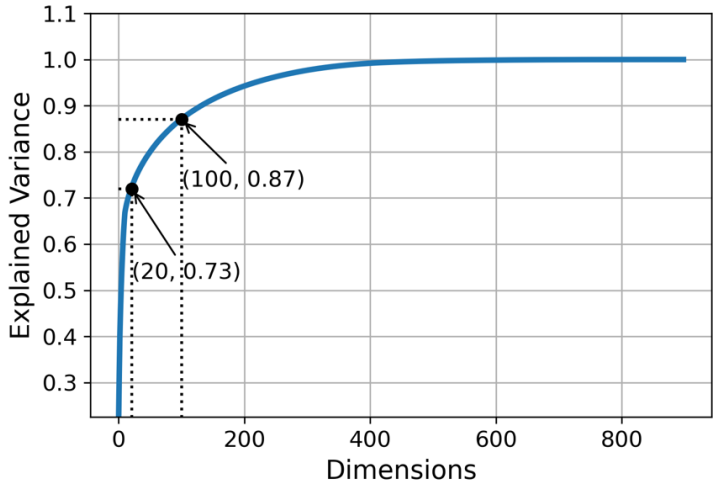


Figure 4.1 PCA Explained Variances with different Dimensions

In terms of the dimensionality reduction of the prediction target variable, the burnt area, $P_1 \cup P_2 \cup P_3$ are used as the training set and $P_4 \cup P_5$ are used for evaluating the models' performance. In addition, the number of training epochs for the CAE models is fixed as 1000. The reconstructed MSE results for different models with different latent space dimensions are shown in Table 4.1. Regarding the compressing of climatic conditions T, H, R *and* L, 5/6 data is used for PCA and CAE model training, and the rest of the data

is used for model testing. Furthermore, comparing reconstruction errors, PCA and CAE perform similarly when compressing climate variables.

Table 4.1 PCA and CAE performance

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

According to Table 4.1, for both PCA and CAE with 100-dimensional latent space, the reconstruction error is lower compared to 20-dimensional. Furthermore, it can be evaluated from the experimental results that the CAE reconstruction MSE is lower compared to PCA. As previously described, CAE can capture feature information in 2-dimensional images during the encoding process, while PCA, which performs compression in 1-dimension space, may ignore these spatial patterns. In summary, CAE with 100-dimensional latent space has the best reconstruction performance in experiments.

4.3 Surrogate model and Latent Data Assimilation

After completing the ROM, the original JULES-INFERNO data could be compressed to the specific latent space. Prior to implementing the surrogate model, the climate and fire data from JULES-INFERNO are encoded by the PCA or CAE models, which could reduce the parameters in later predictions. Furthermore, the fire and four climate variables should be combined as inputs to the surrogate model for achieving the same inputs as the original physical model. The combined data involving P_1, P_2 and P_3 were used as the training set and those containing P_4 and P_5 were used as the test set.

Different ROMs (mentioned in the previous section) compress the JULES-INFERNO data in this project. The encoded data obtained from different ROMs were then used to train the prediction model (LSTM) separately. Additionally, each prediction model has trained 10000 epochs. This research uses the iterative prediction for surrogate models' evaluation, i.e., input the first year (12 snapshots) data and then use the current predicted result for next-level prediction, until the 30^{th} year's fire condition is predicted. The MSE for different surrogate models are shown in Table 4.2.

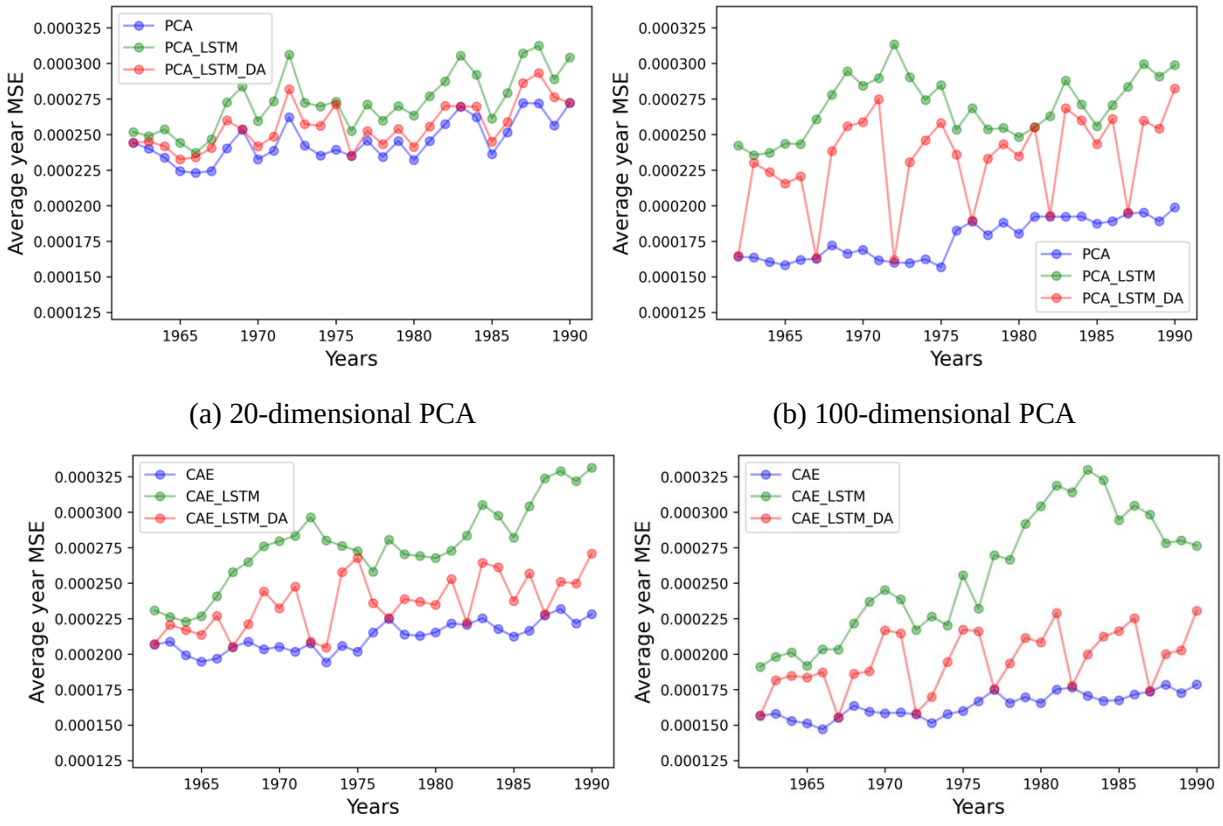
Table 4.2 Surrogate Models' performance

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

Table 4.2 shows that the predictions obtained using PCA down to 20 and 100 dimensions and CAE reduced to 20 dimensions are similar, with an MSE of close to 0.000273. The best performance in LSTM prediction is obtained by applying CAE with 100-dimensional latent space, with an MSE loss of approximately 0.00025.

Additionally, there is a gap between the MSE of the prediction model and the ROM reconstruction, mainly because of the accumulation of errors during the iterative prediction process. Therefore, the project periodically used the observations (raw data) and LA to adjust the forecast results to stabilize subsequent forecasts. LA is implemented every five years during the prediction phase in this study. According to Table 4.2, there is a steady reduction in forecast MSE after correction by LA. The most significant reduction of MSE is obtained with 100-dimensional CAE compared to other approaches.

To better evaluate the surrogate models, we display the MSE of the prediction against time in different surrogate models. In this study, the existing climate and fire data in the 1961 year (provided by JULES-INFERNO) is given to predict the subsequent 29 years of fire. Figure 4.2 shows the MSE per prediction for the different surrogate models evaluated on P_4 test set and plots the ROMs losses for comparison.



(a) 20-dimensional PCA

(b) 100-dimensional PCA

(c) 20-dimensional CAE

(d) 100-dimensional CAE

Figure 4.2 Evolution of MSE against the years in different Surrogate models

According to Figure 4.2, it can be seen that using LA can effectively reduce the prediction error, as the vast majority of red points are lower than the corresponding green points. Furthermore, using CAE-based surrogate models can adapt to LA, stabilizing the predictions afterwards and reducing the accumulation of errors. While applying the LA to the PCA-based surrogate models reduces the error after the simulation, there may still be a sharply increasing forecast error in the subsequent year, as seen in Figure 4.2 (a, b). On the other hand, CAE-based models manage to maintain the improvement of DA for future time steps, as shown in Figure 4.2 (c, d). These results demonstrate the advantage of CAE compared to PCA in terms of generalizability when applied to unseen data. This effect has also been highlighted in the work of [12]. Comparing the four cases presented in Figure 4.2, the 100-dimensional CAE demonstrates the strength of both the original prediction and the assimilated one.

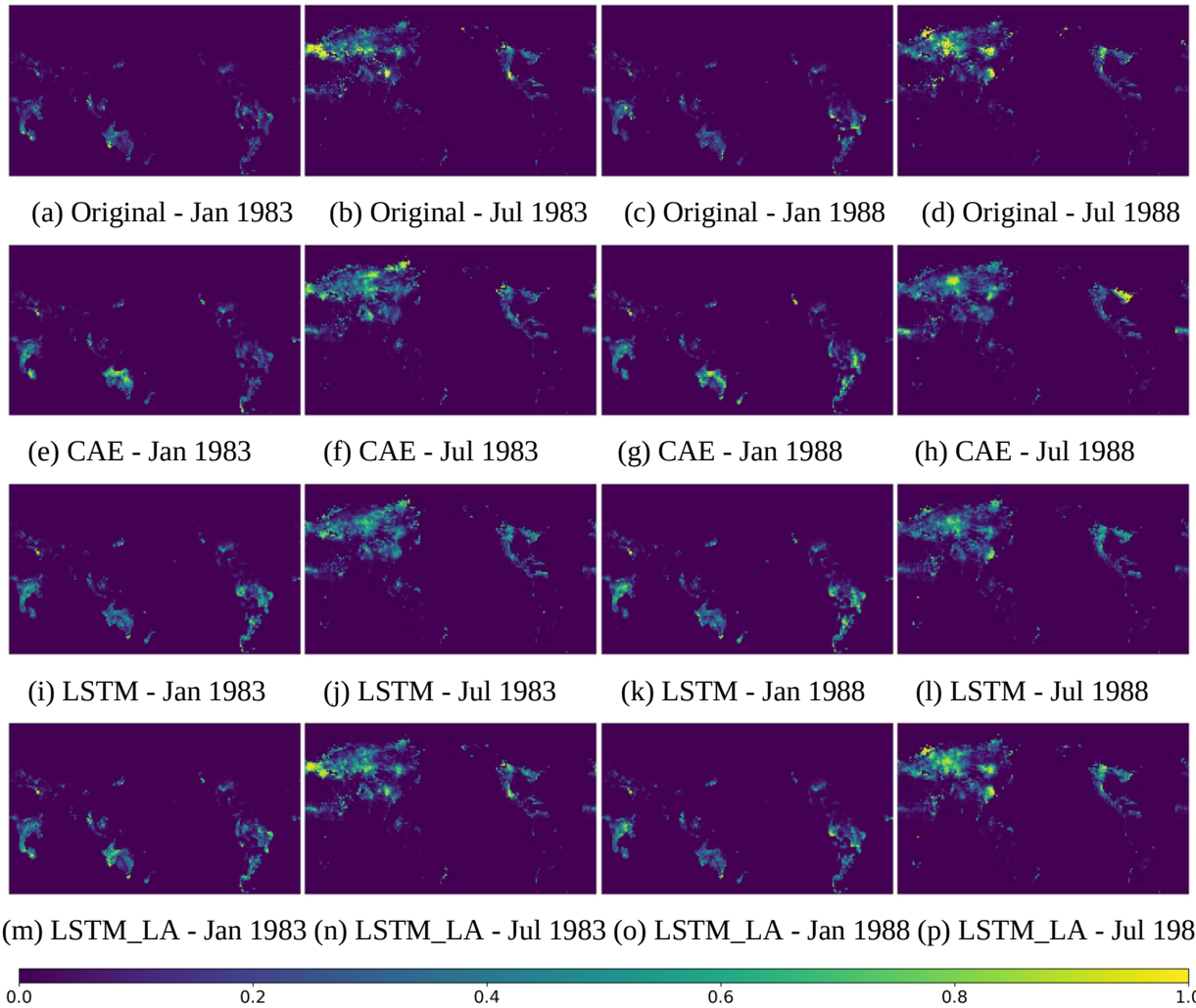


Figure 4.3 The nomadized output of 100-dimensional CAE-based Surrogate Model: the assimilated results (m, n, o, p) are compared with the original simulation (a, b, c, d), the CAE reconstruction (e, f, g, h), and the image predicted by the LSTM model (i, j, k, l)

Figure 4.3 displays the output of the100-dimensional CAE-based surrogate model of year 1983 and 1988 (one year after LA is implemented), and the data is normalized. The horizontal and vertical coordinates in

Figure 4.3 represent latitude and longitude, and the color values in the figure represent the grid box mean burnt area fraction. The brighter the color, the larger the value, color bar is shown in the bottom of Figure 4.3. In addition, it can be seen from Figure 4.3 that forest wildfires in January are mainly in the southern hemisphere, such as Oceania in (a) and South America in (c). On the other hand, the forest fires in July are mainly in the northern hemisphere, such as continental Europe in Figure 4.3 (b, d).

As observed in Figure 4.3, the CAE can effectively reconstruct most of the features in the original image. Furthermore, comparing the LSTM predicted images (i, j, k, l) and the optimisation results after applying LA (m, n, o, p), it is found that the assimilation results are closer to the original image.

Overall, the research implements four ROMs based on PCA and CAE, and the comparison reveals that the 100-dimensional CAE has the best reconstruction performance. Then, after combining the ROMs with prediction models (LSTM), it was found that there was a gap between the prediction results and the ROM reconstruction results. Eventually, the predictions of the surrogate model were made more stable after regular application of LA to adjust the prediction results.

Chapter 5
Conclusion and Future work

5.1 Conclusion

In conclusion, a Deep Neural Network approach was implemented in this study to replace the physical wildfire model JULES-INFERNO, whilst using the same requirements of input and output. Specifically, the surrogate model builds ROMs to encode the data into latent space to reduce subsequent processes' computational costs. Then it builds and uses the compressed data to train LSTM network. Finally, it applies LA to stabilise the prediction results when using the prediction model for iterative prediction to avoid error accumulation. According to the results, the constructed CAE reconstruction is better than PCA. However, as for the iterative prediction, the errors accumulate over time. Applying LA periodically to optimise the prediction results can address this issue and achieve stable long-term predictions. Ultimately the research achieved a more efficient wildfire-prediction surrogate model based on JULES-INFERNO.

5.2 Future work

Currently, all the data for this project is focused on a fixed 30-year period (1961-1990), and no actual wildfire predictions have been made. In the future, more real wildfire data can be applied to evaluate and revise the model to achieve more targeted and accurate predictions. In addition, the surrogate model contains ROMs, predictive models, and LA that could be applied in isolation from this wildfire data. This is because the input and output dimensions in the model are all tuneable. Therefore, this surrogate model can be used in the future with other datasets to achieve predictions in different domains.

References

[1] A. Aldersley, J. S. Murray and E. S. Cornell, "Global and regional analysis of climate and human drivers of wildfire," *Science of the Total Environment*, vol. 409, no. 18, pp. 3472-3481, 2011.

[2] C. Pais, A. Miranda, J. Carrasco and Z. M. Shen, "Deep fire topology: Understanding the role of landscape spatial patterns in wildfire occurrence using artificial intelligence," *Environ Model Software*, vol. 143, pp. 105-122, 2021.

[3] S. Marques, M. Marto and V. Bushenkov, "Addressing wildfire risk in forest management planning with multiple criteria decision-making methods," *Sustainability*, vol. 9, pp. 298, 2017.

[4] M. Claussen, et al., "Earth system models of intermediate complexity: closing the gap in the spectrum of climate system models," *Climate Dynamics*, vol. 18, pp. 579–586, 2002.

[5] I. C. Prentice and S. A. Cowling, "Dynamic global vegetation models," *In S. A. Levin (Ed.), Encyclopedia of biodiversity*, vol. 2, pp. 670-689, 2013.

[6] C. Burton, et al, "Representation of fire, land-use change and vegetation dynamics in the Joint UK Land Environment Simulator vn4.9 (JULES)," *Geosci. Model Dev.*, vol. 12, pp. 179–193, 2019.

[7] S. Mangeon, et al., "INFERNO: a fire and emissions scheme for the UK Met Office's Unified Model," *Geosci. Model Dev.*, vol. 9, pp. 2685–2700, 2016.

[8] K. Nadeem, S. W. Taylor, D. G. Woolford, and C. B. Dean, "Mesoscale spatiotemporal predictive models of daily human- and lightning-caused wildland fire occurrence in British Columbia," I*nternational Journal of Wildland Fire*, vol. 29, no. 1, pp.11, 2020.

[9] A.T. Mohan and D. V. Gaitonde, "A deep learning-based approach to reduced order modeling for turbulent flow control using LSTM neural networks", *arXiv preprint arXiv:1804.09269*, 2018.

[10] H. Gong, Y. Yu, Q. Li, and C. Quan, "An inverse-distance-based fitting term for 3D-Var data assimilation in nuclear core simulation," *Annals of Nuclear Energy*, vol. 141, 2020.

[11] X. Ma, X. Lu, Y. Yu, J. Zhu and J. Chen, "Assimilation of ensemble-variable hybrid data in numerical weather forecasting and its research progress," Journal of Tropical Meteorology, vol. 20, no. 6, pp. 1188-1195, 2014.

[12] M. Peyron, et al., "Latent space data assimilation by using deep learning," *arXiv preprint arXiv:2104.00430*, 2021.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[14] P. Ciais, et al., "Carbon and Other Biogeochemical Cycles," *Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, pp. 465-570, 2014.

[15] O. Pechony and D.T. Shindell, "Fire parameterization on a global scale," *J. Geophys. Res.*, vol. 114, 2009.

[16] M. Kasoar, personal correspondence, https://imperiallondon-my.sharepoint.com/personal/mk1812_ic_ac_uk/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fmk1812%5Fic%5Fac%5Fuk%2FDocuments%2FData%2FJULES%2DINFERNO, 2022.

[17] K. Khowaja and M. Shcherbatyy, "Wolfgang Karl Härdle Surrogate Models for Optimization of Dynamical Systems," *arXiv*, 2021.

[18] F. M. Blanchi, et al., "Short-term electric load forecasting using echo state networked and PCA decompensation," *IEEE*, vol. 3, pp. 1931-1943, 2015.

[19] Z. Huang, et al., "Unsupervised domain adaptation for speech emotion recognitions using PCANet," Multimedia WoIs and applications, vol 76, no. 5, pp. 6785-6799, 2017.

[20] J. Mascl, et al., "Stacked convolutional auto-encoders for hierarchical feature extraction," *International Conference on Artificial Neural Networks. Heidelberg: Springer*, 2011.

[21] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.
[22] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
[23] JJ. Vallino, “Improving marine ecosystem models: Use of data assimilation and mesocosm experiments,” *Journal of Marine Research*, vol. 58, pp. 117-164, 2000.
[24] R. N. Bannister, “A review of operational methods of variational and ensemble-variational data assimilation,” *Q. J. R. Meteorol*, vol, 143, 607–633, 2017.
[25] S. Cheng, et al, “Error covariance tuning in variational data assimilation: application to an operating hydrological model,” *Stoch Environ Res Risk Assess*, vol. 35, pp. 1019–1038, 2021.
[26] O. A, Stepanov, "Kalman filtering: Past and present. An outlook from Russia. (On the occasion of the 80th birthday of Rudolf Emil Kalman),” *Gyroscopy and Navigation*, vol. 2, no. 2, p. 105, 2011.
[27] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen, “Data assimilation in the geosciences: An overview of methods, issues, and perspectives,” *Wiley Interdisciplinary Reviews: Climate Change*, vol. 9, no. 5, pp. 535, 2018.
[28] A. S. Lawless, S. Gratton, and N. Nichols, “Approximate iterative methods for variational data assimilation,” *International journal for numerical methods in fluids*, vol. 47 no. 10-11, pp. 1129–1135, 2005.
[29] W. Fulton, “Eigenvalues, invariant factors, highest weights, and Schubert calculus,” *Bulletin of The American Mathematical Society*, vol. 37, pp. 209–250, 2000.
[30] M. Luo, X. HUANG and S. LU, “Urban gas daily load forecasting based on PCA-LSTM,” *Proceedings of China Gas Operation and Safety Seminar (the 10th) and 2019 Aca- demic Annual Meeting of Gas Branch of China Civil Engineering Society*, pp. 120 – 132, 2019.

Appendix A.

The principal of PCA

The compressed variable $\tilde{u}_i$ can be represented by a linear combination of $u_1, u_2, \ldots, u_n$ as shown in Equation (10), to maximise the global variance after the transformation.

$$\begin{cases} \tilde{u}_1 = \alpha_{11}u_1 + \alpha_{12}u_2 + \cdots + \alpha_{1n}u_n \\ \tilde{u}_2 = \alpha_{21}u_1 + \alpha_{22}u_2 + \cdots + \alpha_{2n}u_n \\ \tilde{u}_3 = \alpha_{31}u_1 + \alpha_{32}u_2 + \cdots + \alpha_{3n}u_n \\ \vdots \\ \tilde{u}_k = \alpha_{k1}u_1 + \alpha_{k2}u_2 + \cdots + \alpha_{kn}u_n \end{cases} \tag{10}$$

where, $\alpha_i = [\alpha_{i1}, \alpha_{i1}, \cdots, \alpha_{in}]^T$ $(i = 1, 2, \cdots, k)$ is the eigenvector associated to the eigenvalues λ_i of the covariance matrix C. The covariance matrix C is shown in Equation (11), where the formula for the covariance is shown in Equation (12).

$$C = \begin{bmatrix} cov(u_1, u_1) & cov(u_1, u_2) & \cdots & cov(u_1, u_n) \\ \vdots & & \ddots & \vdots \\ cov(u_n, u_1) & cov(u_n, u_2) & \cdots & cov(u_n, u_n) \end{bmatrix} \tag{11}$$

$$cov(u_i, u_j) = E(u_i u_j) - E(u_i)E(u_j) \quad i, j = 1, 2, \cdots, n \tag{12}$$

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

The contribution rate of each principal component can be calculated using the principal component explained variance formula (Equation (13)), the principal components corresponding to the special eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$were selected in the descending order to construct $\widetilde{U}$. The cumulative explained variance is referred to in Equation (14).

$$\alpha_i = \frac{\lambda_i}{\sum_{j=1}^{k} \lambda_j}, i, j = 1, 2, \cdots, k \tag{13}$$

$$\sum_{i=1}^{k} \alpha_i = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{k} \lambda_j}, i, j = 1, 2, \cdots, k \tag{14}$$

The principal of LSTM

The essence of the gate structure of the LSTM is to use the sigmoid as the activation function so that the fully connected network layer outputs a value between 0 and 1, describing the proportion of the information quantity passed. The forgetting gate indicates the proportion of the output information quantity forgotten at the last moment, and the input gate represents the proportion of the input information quantity retained at the current moment, both updating the state value together. Finally, the output gate represents the proportion of the new state output. Suppose x_t $(x_0 = [Z_{s,i}, Z_{s,i+1}, \cdots, Z_{s,i+t_{in}-1}], s = 1,2,3, i = 1,2, \cdots, 349)$ is the current input, h_{t-1} is the output of last moment, c_t is the value of new state, $c_{t-1}$is the state value of last moment. The Equations (15, 16, 17, 18, 19, 20) represent the inner principal of the LSTM.

Input state:

$$z_t = \tanh(W_z x_t + Q_z h_{t-1} + b_z) \tag{15}$$

Input gate:

$$i_t = \sigma(W_i x_t + Q_i h_{t-1} + b_i) \tag{16}$$

Forget gate:

$$f_t = \sigma\left(W_f x_t + Q_f h_{t-1} + b_f\right) \tag{17}$$

Current state:

$$c_t = f_t * c_{t-1} + i_t * z_t \tag{18}$$

Output gate:

$$o_t = \sigma(W_o x_t + Q_o h_{t-1} + b_o) \tag{19}$$

Current output:

$$h_t = o_t * \tanh(c_t) \tag{20}$$

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