Unveiling the complex dynamics of earthquakes through spatial-temporal data analysis

Bobba Upendar Reddy[a], **Chung Jun Yu Chester**[a], **and Zhong Wei Liang**[a]

[a]College of Design and Engineering, National University of Singapore, Singapore

This manuscript was compiled on April 10, 2023

Earthquakes can be described as a highly non-linear and complex dynamical system, without a specific mathematical model governing it. This study was designed to extract the main spatial and temporal dynamics of the system, based on methods such as Fast Fourier Transform, Gabor Transform and Dynamic Mode Decomposition. Another aim of the study was to investigate the predictability of the system, by identifying the presence of chaos using the Lyapunov exponent and by performing future state prediction using neural networks. Results indicate that the occurrence of earthquake seem to follow a harmonic pattern and tend to cluster at specific spatial plots. While short-term prediction of future states can provide useful insight into the dynamics and complexity of earthquake occurrences, long-term behaviour of the system is extremely sensitive to past events and can therefore be challenging to predict.

Earthquakes | Spatial-temporal data | Lyapunov Exponent | Dynamic Mode Decomposition | Neural Networks

1. Introduction

Earthquakes are natural phenomena that occur due to tectonic plate movements modelled as surfaces which undergo slip and can cause widespread destruction and loss of life. These phenomena take place on faults, which are zones of fractures between two blocks of rock. The existence of faults enables the blocks to move relative to each other and if this movement occurs rapidly, an earthquake is generated.

Understanding the spatial-temporal patterns of earthquake occurrences is crucial in possibly predicting future events and potentially mitigating their effects. The aim of this project is to mainly analyse the spatial-temporal data of earthquake occurrences and identify any patterns that may exist. The main quantity of interest which is crucial in analysing the dynamics of earthquakes is the slip potency rate, which is referred to as the product between the average slip rate and the area that experienced the slip as follows:

$$Slip\ potency\ rate\ (m^3/s) = Average\ slip\ rate\ (m/s) \times Area\ of\ slip\ (m^2) \tag{1}$$

The average slip rate $(units : m/s)$ refers to the average velocity at which two sides of a fault traverse relative to each other during the occurrence of an earthquake. The area of slip $(units : m^2)$ refers to the surface area of the fault plane that underwent slip during the occurrence of an earthquake. Simply, the slip potency rate $(units : m^3/s)$ is a measure of the amount of energy released during an earthquake, and is commonly used as a metric to approximate the seismic hazard of a region.

To investigate the dynamics of earthquakes, we study a spatial-temporal dataset which is designed to imitate the simulation of complex earthquake occurrences on a particular planar fault. To describe the dataset broadly, the dataset is divided spatially into 20 × 128 patches, with each patch having an equal size of 2.5 × 2.5 km^2. Specifically, the number of latitudes is 20 and the number of longitudes is 128, and each spatial patch can be described in terms of their individual latitudinal and longitudinal values. The slip potency rates for each patch are provided in the dataset. Furthermore, the dataset consists of a total of 56825 sampled epochs, with a sampling rate of 1 data point per day. Therefore, the dataset to be analysed is a 3-dimensional spatial-temporal dataset.

To properly investigate earthquake dynamics, it is essential to first understand the variations in slip potency rates over time. Figure 1 shows the slip potency rate variation along the latitudes and longitudes for different sampled times or epochs. The yellow colour indicates a high slip potency rate, a turquoise colour indicates a moderate slip potency rate, while dark blue indicates a low slip potency rate close to zero. The plot seemingly shows that slip potency rates do not follow any patterns or trends over time, suggesting that earthquake occurrences are complex and dynamic. analysing the spatial-temporal dataset of earthquake occurrences sheds light on the complexity of these events. The slip potency rates show significant spatial and temporal variations, with no clear patterns or trends.

In this study, we utilise several tools to extract information from the dataset, such as Fast Fourier Transform (FFT) to analyse the temporal patterns of the system, computing the Lyapunov exponent to detect the presence of chaos within the system, and Dynamic Mode Decomposition (DMD) for deriving a linear mapping for the dynamics of the reduced-order model

All authors designed the study, analysed the data, drafted the manuscript, and revised it critically for important intellectual content. All authors approved the final version of the manuscript for submission.

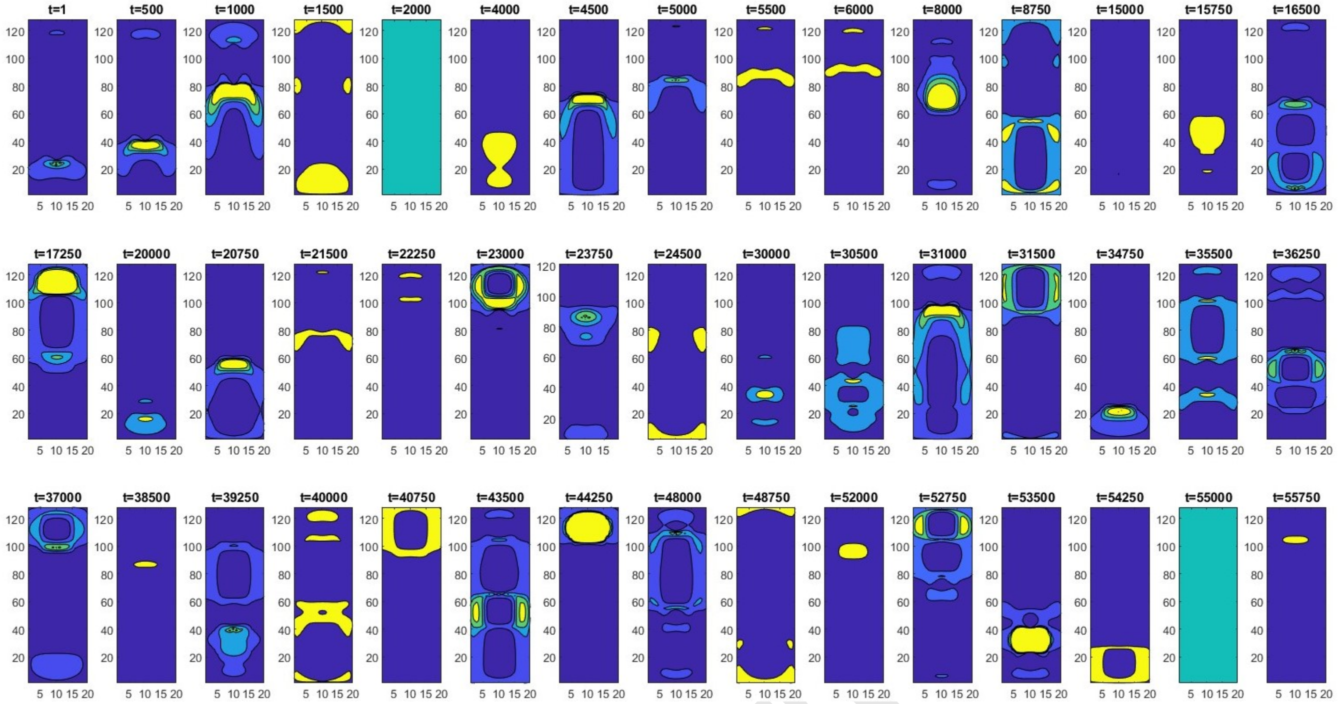


Fig. 1. Evolution of system. Variation of slip potency rates over time. The x-axis represents the latitudes and the y-axis represents the longitudes of the spatial plots. Colours of the plots represent the magnitude of the slip potency rates, with yellow, turquoise and dark blue indicating high, moderate and low slip potency rates respectively.

of the system for possible future state prediction. Our goal is to characterize the dynamics of the system and to investigate whether the system can be accurately predicted or not.

2. Temporal analysis of system

2.1. Importance of temporal analysis.

Temporal analysis of a system involves studying and analysing the behavior of the system over time. It is a commonly used technique that enables researchers to understand how a system evolves and changes over a certain period of time. This analysis is especially fundamental in fields such as engineering and physics, where it is often crucial to predict how a system will behave in the future based on its past and current behavior. Temporal analysis typically involves gathering data on the system at different points in time and analysing how the system changes over those time periods. This entails observing trends in the data, identifying certain recurring patterns or cycles, and modelling the system's behavior using various statistical or mathematical techniques. In this section, we focus on two different techniques to extract the important temporal dynamics that describe the system of earthquakes, namely the Fast Fourier Transform (FFT) and the Gabor Transform.

2.2. Fast Fourier Transform.

FFT is a widely-used mathematical algorithm (1) for efficiently computing the Discrete Fourier Transform (DFT) of a discrete signal. The DFT is a method of analysing a time-domain signal and converting it into its frequency-domain representation (2). The DFT decomposes a finite-length sequence of discrete samples, into its corresponding sinusoidal components of different frequencies. Each frequency component is described by a complex number, whereby the magnitude refers to the amplitude of the sinusoid and the phase represents its relative position in time. The DFT is defined by the following equation:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}} \qquad [2]$$

In Equation 2, $X(k)$ refers to the k-th frequency component of the signal, $x(n)$ refers to the n-th sample of the time-domain signal, and N is the length of the signal. Naive computation of the DFT requires $O(N^2)$ operations, which most often becomes computationally expensive for large signals. Therefore, the FFT is an algorithm that tackles this problem by decreasing the computational complexity of the DFT to $O(N * log(N))$, resulting in a much faster and more efficient computation for large signals. The fundamental workings of the FFT is to decompose the signal into smaller sub-signals and then apply the DFT recursively to each individual sub-signal. Utilising a butterfly operation (3), the sub-signals are then merged to generate the final frequency-domain representation of the signal.

Epochs	Highest slip potency rate	Longitude of earthquake epicentre (x)	Latitude of earthquake epicentre (y)
1	0.0905	10	24
2	0.0909	10	24
3	0.0912	10	24
4	0.0915	10	24
⋮	⋮	⋮	⋮
56825	0.2531	99	10

Table 1. Table containing the highest slip potency rate, longitude and latitude of the earthquake epicentres for each epoch

To form the raw signal, we took the simple approach of obtaining the highest slip potency rate for each epoch, as described in Table 1. Figure 2a describes the plots of slip potency rates against the sampled epochs for the raw and filtered signals. The graph is plotted with limits on the vertical axis (slip potency rate) from -0.2 to 1.0, in order to aid in visualization of the plots. The filtering of the raw signal to obtain the filtered signal is carried out in a series of steps. Firstly, FFT is applied on the noisy signal such that: $\mathcal{F}[f_{noise}(t)] = \hat{f}_{noise}(\omega)$. The power spectral density (PSD) is then computed by: $PSD =< \hat{f}, conj(\hat{f}) >$. A threshold value for the PSD is then set at a relatively high value. In our analysis, we determined $PSD = 105$ as a good threshold value to filter the signal sufficiently well. All indices corresponding to the PSD above the threshold value are retained, while the rest of the indices that correspond to the PSD below the threshold value are set to zero and truncated. Effectively, this filters out the noise within the raw signal that is associated to the lower peaks in the spectrum, and enables identification of high peaks in the spectrum, which corresponds to the frequency of the filtered signal, as depicted in Figure 2b.

From Figure 2a, it is evident that the filtered signal generally displays a harmonic motion throughout the time frame. In fact, the dominant pattern in the filtered signal is that the amplitude of the signal seems to increase and decrease repeatedly, which indicates the amplification and damping cycles of the signal. In the context of earthquake data, these cycles can be described by the accumulation and attenuation phases of the seismic waves before and after the release of energy during an earthquake occurrence, which is further explained in the subsection Gabor Transform. Figure 2b illustrates the FFT plots for the raw and filtered signals. We see that the dominant frequency of the filtered signal equates to 0.000263964 samples/day, which means that the oscillation period of the signal is $T = \frac{1}{0.000263964} \approx 3788$ days. Hence, this indicates, not guarantees, that there could be a sudden burst of energy which could result in an earthquake event every 3788 days approximately. However, from the FFT plot alone, we are unable to deduce the approximated slip potency rates of those events, as well as their estimated location of occurrence. Nevertheless, temporal analysis through FFT enables us to broadly study the temporal dynamics of the signal to identify its representative frequencies.

2.3. Gabor Transform.

The Gabor transform is a mathematical technique commonly utilised in image analysis and signal processing to analyse the frequency content of an image or signal in both time and frequency domains. The Gabor transform involves multiplying a signal or image by a window function that is modulated by a sinusoidal wave (4). The window function, usually a Gaussian function, aids in localizing the transform in both time and frequency domains. The resulting transformed image or signal contains information about the amplitude and phase of the frequency components of the original image or signal. Therefore, this makes the Gabor transform a useful tool for analysing signals that consist of both time-varying and frequency-varying components.

To analyse the temporal dynamics of our spatial-temporal data set, we utilise the Gabor transform to produce a spectrogram, which is a visualization of the frequency content of a particular signal over time. To generate the spectrogram, the signal is first divided into short windows or segments, and the Gabor transform is applied to each window. The resulting transformed signals consist of information about the frequency content of each window, and these can be used to construct the spectrogram by plotting the magnitude or power of each frequency component as a function of time. In our study, the signal is formulated by taking the maximum slip potency rate at each epoch for the entire time frame (56825 epochs), as shown in Table 1.

Figure 2c illustrates the spectrogram of the raw signal, derived by Gabor transform and plotted using the MATLAB function *spectrogram*. We designed the spectrogram such that it divides the raw signal into segments of length (365 days) and windows each segment with a Hamming window of that length. Furthermore, the number of overlapped samples is set at 292 samples, which equates to 80% of the window (365 days). In addition, the number of DFT points is fixed at 365, and the sampling frequency $fs = 1.157 \times 10^{-5}$ Hz (1 data point per day) is also accounted for.

From Figure 2c, it is evident that the spectrogram plot is populated with numerous peaks with varying degrees of intensities or brightness. As the intensity of the image describes the magnitude of the signal at each time-frequency point, this indicates that the bright signals represent regions where the seismic waves are particularly pronounced or strong. Upon closer inspection, we see that each peak is mostly accompanied by fading levels of intensity before and after it. This highlights that the process of an earthquake occurrence can be broadly described in three phases. The first phase refers to the accumulation of the seismic waves, which results in a gradual build-up of energy. The second phase takes place when there is a sudden burst or release of this energy, resulting in an earthquake. The energy of the signal is distributed across a range of frequencies rather than being concentrated at a single frequency, which explains the vertical bands depicted on the spectrogram. The third phase refers to the attenuation of the seismic waves as they scatter and propagate away from the epicentre through the Earth, resulting in a gradual reduction in the energy of the waves.

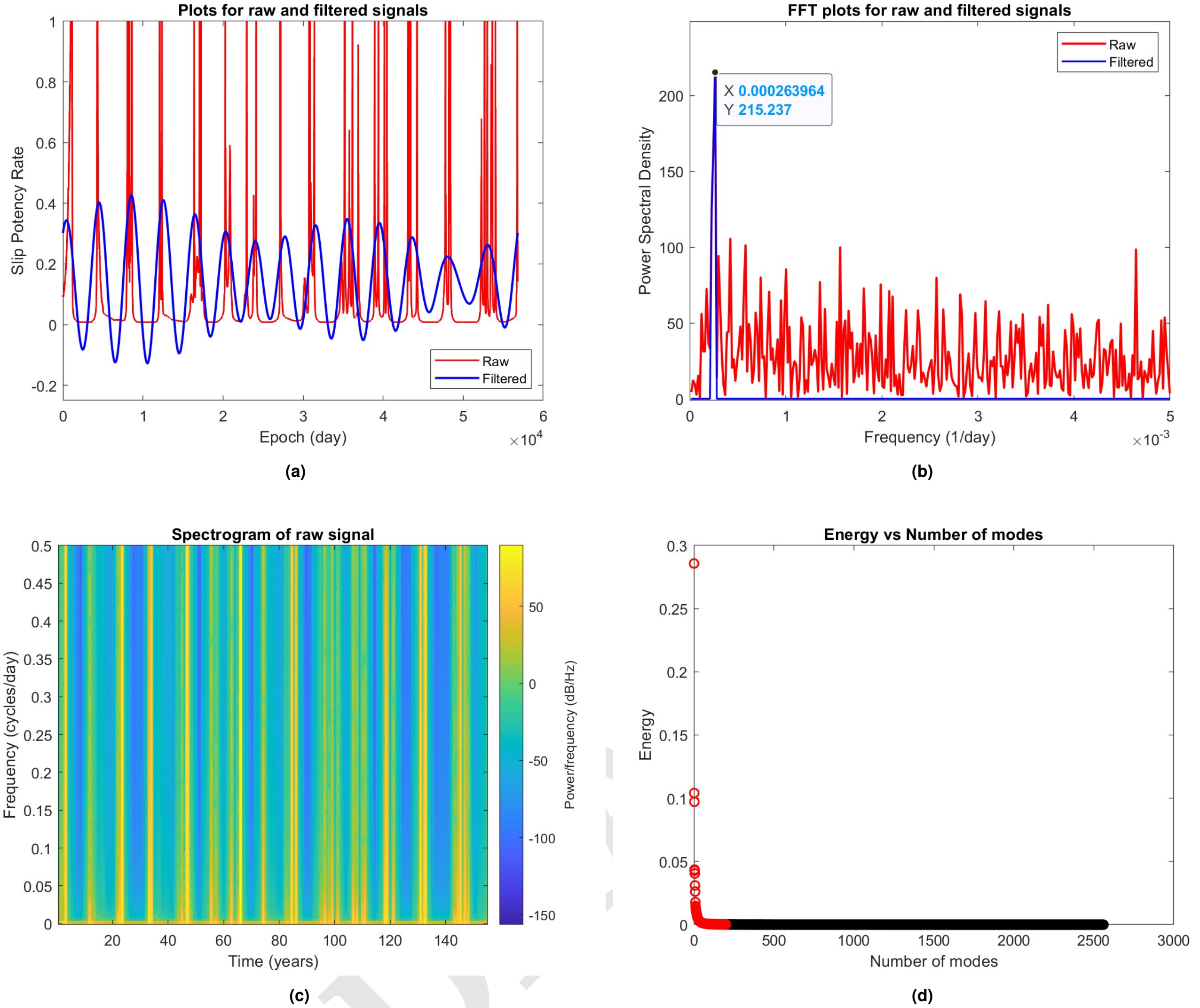


Fig. 2. (a) Plots for raw and filtered signals. The filtering process consists of truncating the frequencies with power spectral density below a threshold value. Small Fourier coefficients are also negated. **(b)** FFT plots for the raw and filtered signals. **(c)** Spectrogram of the raw signal. The x-axis represents the time in years, while the y-axis represents the frequencies of the components within the signal. The colour or intensity of the image describes the magnitude or power of the signal at each time-frequency point. **(d)** SVD Truncation. Plot of Energy against number of modes. Total number of modes is 128 $\times$ 20 = 2560 modes. 200 modes (denoted by red circles) are chosen to construct the reduced-order model.

One important point to note is that not all the vertical bands in the spectrogram necessarily indicate an occurrence of an earthquake. Some of them could be caused by noise or interference in the signal, and it is therefore essential to carefully analyse the surrounding data to properly determine the significance and cause of the vertical bands forming.

3. Reduced order modelling

3.1. Complexity of high-dimensional dynamical systems.

For high-dimensional dynamical systems with several degrees of freedom, it can be computationally complex and expensive to derive solutions for the partial differential equations (PDEs) of these systems. Pertaining to the spatial-temporal data of real-world phenomena such as earthquakes in this study which is classified as a system with many degrees of freedom, we wish to identify a reduced-order model (ROM) that is still able to capture the underlying dynamics of the original system with a great degree of accuracy. The main purpose of deriving a low dimensional manifold of our data is to significantly reduce computational costs, which allows for real-time prediction of future states of the data set. In fact, an increasing number of parametric model order reduction (MOR) techniques and data-driven methods for reduced-order modelling have emerged in recent years to handle progressively complex computational tasks (5).

3.2. Singular Value Decomposition.

Singular Value Decomposition (SVD) is a popular technique utilised mainly for data compression and dimensionality reduction purposes in linear algebra and has many important applications in fields such as signal processing and data analysis. SVD extracts the correlated dominant features from a given data matrix by decomposing the matrix into three separate matrices:

$$U \in \mathbb{C}^{n\times n}$$
$$\Sigma \in \mathbb{R}^{n\times m}$$
$$V \in \mathbb{C}^{m\times m}$$

U is a unitary left singular matrix, Σ is a diagonal matrix of singular values, and V is a unitary right singular matrix. Mathematically, the SVD of matrix $X \in \mathbb{C}^{n\times m}$ is represented as follows:

$$X = U\Sigma V^T \tag*{[3]}$$

U consists of the modes that best represent the spatial correlation of the data matrix X, with the first few modes being the ones that optimally describe the variance in the data. Σ is a diagonal matrix with non-negative elements called singular values. The diagonal entries of Σ are arranged in decreasing order of magnitude such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k \geq 0$, where k takes the minimum value of n (number of rows in data matrix X) and m (number of columns in data matrix X). By performing the SVD, the data matrix X is then estimated with a principled low-rank approximation, that is proven to give the optimal representation of the data in an L_2-sense (6).

To compute the SVD of our spatial-temporal data matrix, the two-dimensional spatial data (consisting of the latitudinal and longitudinal values of the spatial plots) was reshaped into a single column vector. utilising the MATLAB function *svd*, an economy-size decomposition of the data matrix was produced whereby extra rows or columns of zeroes from the diagonal matrix of singular values (Σ), together with the columns in U and V^* that multiply those zeros, were removed. Obtaining the economy-size decomposition does not affect the accuracy of the decomposition, and in fact reduces computational time of the process.

According to the Eckart-Young-Mirsky theorem, the optimal rank-r approximation to data matrix X, in a least-squares sense, is described by the rank-r SVD truncation $\tilde{X}$ as follows:

$$\arg \min_{\tilde{X}, s.t. rank(\tilde{X})=r} ||X - \tilde{X}||_F = \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag*{[4]}$$

$\tilde{U}$ and $\tilde{V}$ describe the first r leading columns of U and V, and $\tilde{\Sigma}$ refers to the r × r sub-block of Σ. $||\cdot||_F$ refers to the Frobenius norm, which is defined as the matrix norm of a matrix given by the square root of the sum of the absolute squares of its elements. Generally, the theorem handles the problem of approximating a matrix by one of lower rank (7), and states that for a given r, there is no approximation that is more optimal than the rank-r SVD in the L_2-sense.

To obtain the truncated SVD for the spatial-temporal earthquake data, a suitable low rank space needs to be identified. This is done by plotting the energy of the singular values contained in Σ, and differentiating between the 'important' modes and the 'unimportant' modes which have negligible energy, as illustrated in Figure 2d. Visually, we see that there are approximately 200 modes with significant energy, which should not be truncated in order to preserve an accurate description of the underlying dynamics of the data. Therefore, by taking the first 200 leading columns of U and V as well as the 200 × 200 sub-block of Σ, we form a reduced-order model that can be described by $\tilde{U}$, $\tilde{V}$ and $\tilde{\Sigma}$.

3.3. Dynamic Mode Decomposition.

Dynamic Mode Decomposition (DMD) is a mathematical technique used to analyse and extract the underlying dynamics of a system based on observations and measured data. As it is a data-driven method that does not require any prior knowledge of the system dynamics, it is a useful tool that can be utilised for analysing complex systems. The main purpose of using DMD is to identify the spatial-temporal coherent structures that are present in the data, and to extract their corresponding temporal evolution. Essentially, the DMD algorithm is a matrix factorization method based on the SVD algorithm, and also contains aspects of Fourier transform in time (6).

The fundamental idea behind DMD is to first collect data from a system over a period of time. For the spatial-temporal data of earthquakes, we collect snapshots of the slip potency rates for the entire 2560 patches of land as the system evolves in time. Previously, we have noted that there are 56825 epochs in total, with a sampling rate of 1 data point per day. The data is then used to construct two matrices **X** and **X'**, where **X** contains the data at a given time, and **X'** consists of the data at the next time step. To preserve an optimal temporal resolution of the data, the **X'** matrix is shifted by one time step from the **X** matrix as follows:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \cdots & \mathbf{x}(t_{56284}) \\ | & | & & | \end{bmatrix} \tag*{[5]}$$

$$\mathbf{X'} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_2) & \mathbf{x}(t_3) & \cdots & \mathbf{x}(t_{56285}) \\ | & | & & | \end{bmatrix} \tag*{[6]}$$

The purpose of the DMD algorithm is to identify the best-fit linear operator **A**, that relates the two snapshot matrices **X** and **X'**.

$$A = \arg\min_{A} ||X' - AX||_F = X'X^{\dagger} \tag{7}$$

However, as with many complex systems, the best-fit linear operator **A** is high-dimensional, and computing it requires high computational time. For our data, **A** is a 2560 × 2560 matrix. Therefore, instead of directly computing **A**, SVD is performed on **X** to obtain the principal modes of the system. These principal modes encapsulate the underlying dynamics of the system and can be utilised to reconstruct the original data or to predict future behavior of the system. Truncation is also performed by taking the rank-r SVD of the data matrix **X**.

$$\mathbf{X} \approx \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^T \tag{8}$$

In Equation 8, $\tilde{\mathbf{U}} \in \mathbb{C}^{n\times r}$, $\tilde{\Sigma} \in \mathbb{C}^{r\times r}$ and $\tilde{\mathbf{V}} \in \mathbb{C}^{m\times r}$, whereby r ≪ m denotes the approximate rank of the data matrix **X**. In our study, based on experimentation, we selected r = 200 out of a possible 2560 modes, to compute the truncated SVD as it sufficiently captures the underlying dynamics of the data. With the truncated SVD, the reduced matrix $\tilde{\mathbf{A}}$ is computed by projecting onto the leading r = 200 modes:

$$\tilde{\mathbf{A}} = \tilde{\mathbf{U}}^T\mathbf{A}\tilde{\mathbf{U}} = \tilde{\mathbf{U}}^T\mathbf{X'}\tilde{\mathbf{V}}\tilde{\Sigma}^{-1} \tag{9}$$

The reduced matrix $\tilde{\mathbf{A}}$ provides a linear model for the dynamics of coefficients derived by SVD. Simply, $\tilde{\mathbf{A}}$ defines a linear mapping between state vectors which are separated by a single time step: $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k$. The reduced matrix $\tilde{\mathbf{A}}$ contains the same non-zero eigenvalues of the full matrix **A**. Therefore, the need to compute the high-dimensional **A** matrix is removed, as we only need to compute the reduced matrix $\tilde{\mathbf{A}}$. For full state reconstruction from the truncated state, we apply the formula: $\mathbf{x} = \tilde{\mathbf{U}}\tilde{\mathbf{x}}$. This can be used for prediction of future states of the system, which is discussed more in the Prediction using DMD sub-section.

The spectral decomposition of the reduced matrix $\tilde{\mathbf{A}}$ are then derived using the formula: $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda$. Λ refers to the diagonal matrix which consists of the DMD eigenvalues corresponding to the eigenvalues of the high-dimensional matrix **A**. **W** refers to the eigenvectors of the reduced matrix $\tilde{\mathbf{A}}$, which provide a coordinate transformation for diagonalising the matrix. Spectral decomposition is the fundamental function of the DMD algorithm, as it extracts spatial-temporal patterns from the data set (8), and decomposes them into a series of modes, each with a specific frequency and spatial structure. To obtain these series of high-dimensional DMD modes, we utilise the eigenvectors of the reduced system **W** and the time-shifted snapshot matrix **X'** to compute Φ as follows:

$$\Phi = \mathbf{X'}\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\mathbf{W} \tag{10}$$

The high-dimensional DMD modes represented by Φ, are the eigenvectors of the high-dimensional matrix **A**, corresponding to the eigenvalues Λ. By computing Φ, temporal analysis of the system can be conducted by plotting the eigenvalues (Ritz values) of the dynamics matrix in the complex plane, as illustrated in Figure 3a. Furthermore, we are able to extract the spatial structures describing the dynamics of the system by plotting the dominant DMD modes.

From Figure 3a, we see a DMD spectrum displaying the real and imaginary parts of the eigenvalues of the system. Ritz values, also known as Ritz eigenvalues, are approximations of the eigenvalues of a matrix obtained by projection onto a subspace. In the Ritz values plot, the unit circle represents the boundary of the region in the complex plane where the eigenvalues of the original matrix lie. The Ritz values, which are approximations of the eigenvalues, are expected to lie within this circle. if the Ritz values lie exactly on the unit circle in the complex plane, it indicates that the temporal dynamics of the system are periodic with a fixed frequency. This is because the unit circle in the complex plane represents a periodic oscillation with a fixed frequency, and the Ritz values are related to the frequencies of the dominant modes of the system. Therefore, if the Ritz values lie exactly on the unit circle, it suggests that the system has a dominant periodic oscillation with a fixed frequency. This is usually seen in the case of vortex shedding in the analysis of a cylinder wake flow, whereby almost all the Ritz values lie on the unit circle, indicating that the snapshots lie on or near an attracting set (9) and that the temporal dynamics of vortex shedding can be described as periodic.

However, in Figure 3a, it is evident that while most of the Ritz values lie within the unit circle, very few of them lie exactly on it. Rather, we see a huge cluster of Ritz values situated at the point where (Re, Im) = (1,0), which indicates the presence of a dominant DMD mode with a frequency of 1 and no growth rate. Therefore, this strongly suggests, not guarantees, that the dominant mode is a harmonic one, which can be described by formation of standing (stationary) waves. Standing waves are formed when waves propagate in opposite directions and interfere constructively, resulting in the creation of a stationary pattern. In the context of our spatial-temporal data, standing waves can arise due to the interference of waves reflected and refracted from several subsurface structures. In fact, this is corroborated by visualization of our raw data in Figures 3c and 3d, which show the constructive interference between two waves of similar amplitudes travelling in opposite directions, causing a sudden release in energy that is manifested as a standing wave. This is a dominant temporal pattern which occurs frequently throughout the entire time frame of the system.

In DMD, a coherent spatial structure refers to a spatial pattern or structure that is observed in the data and is associated with a dominant DMD mode. A coherent structure can be identified by looking at the spatial structure of the associated DMD mode, which represents a pattern that is repeated over space and time. The amplitude of the DMD mode at each spatial

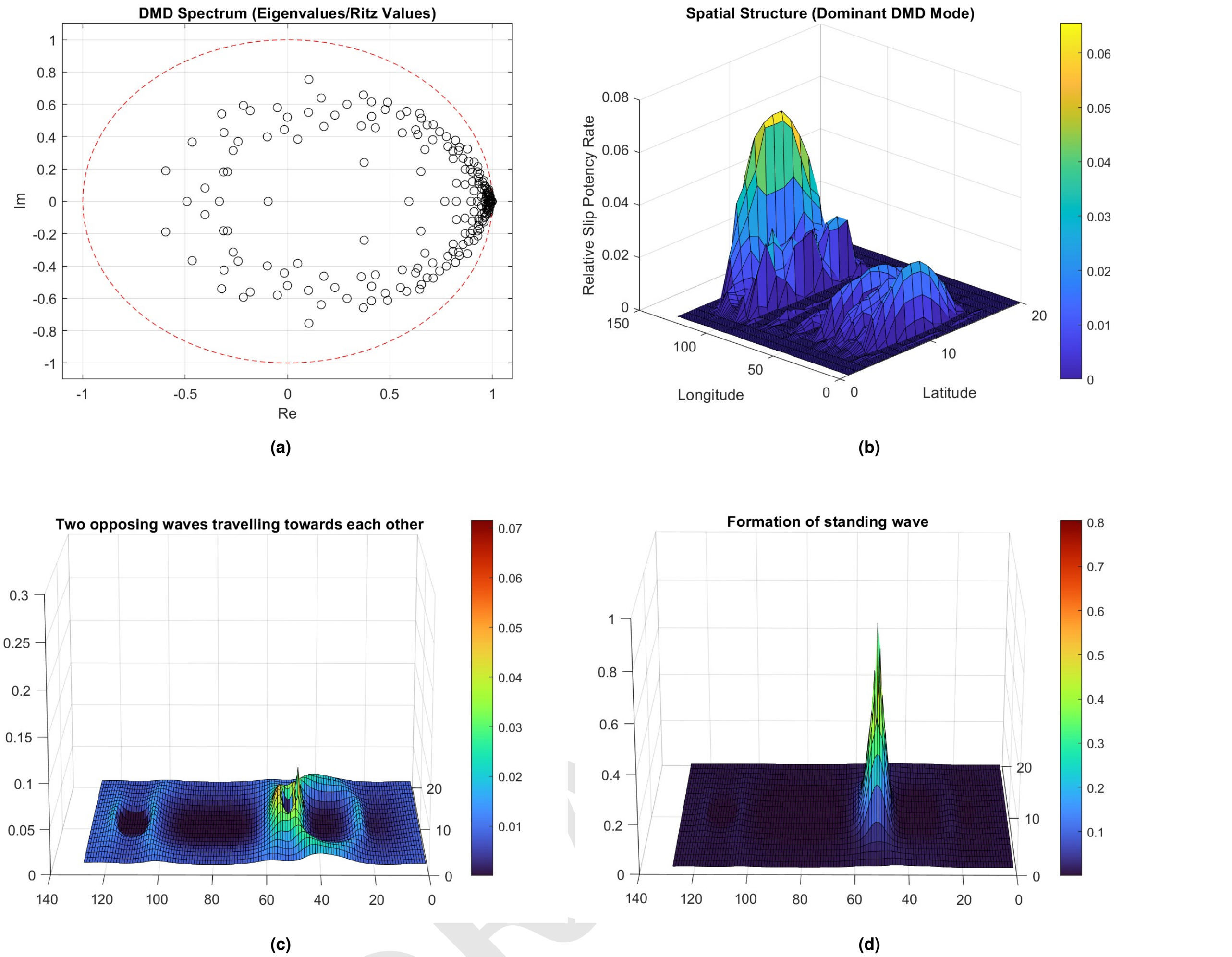


Fig. 3. **(a)** DMD spectrum illustrating the real and imaginary parts of the eigenvalues. Plotted against a red dotted line (unit circle) for comparison. **(b)** A coherent spatial structure that describes the dynamics of the system, as characterized by a dominant DMD mode. We utilise a total of 200 truncated modes for the reduced-order model. **(c)** Illustration of two opposing waves of approximately equal amplitude travelling towards each other, before constructive interference. **(d)** Constructive interference of both waves causes a formation of a standing wave.

location represents the contribution of that location to the overall pattern, and the phase of the mode represents the spatial relationship between the different locations in the pattern. The coherent spatial structures in DMD can provide insight into the underlying dynamics of the system being studied. Figure 3b illustrates one of the coherent spatial structures of the system, whereby there is a prominent standing wave exhibiting relatively high slip potency rates in the region where the longitudinal values range from 110 to 120 and the latitudinal values range from 4 to 16. Through our analysis, we found that most of the coherent spatial structures derived through the dominant DMD modes illustrate a similar spatial pattern. Therefore, this may indicate the presence of a strong heterogeneity or a localized zone of high stress accumulation in the fault zone. As a result, this could be a potential location for future earthquakes as it represents a region of high stress concentration within the fault.

4. Detection of chaos within system

4.1. Chaos Theory.

In modern day, forecasting earthquakes remains a major difficulty, as numerous precursors and methods have been discussed but often lack consistency and reliability in their predictions (10). Despite ongoing efforts to improve earthquake prediction systems, the task still poses a significant challenge due to the complex nature of the phenomenon and the limitations of existing methods. This has led to the common assumption that the occurrence of earthquakes could be a stochastic process rather than a deterministic one. In this section, we investigate the dynamics of the spatial-temporal dataset of the slip potency rates, by utilising chaos theory.

Chaos is the stochastic behaviour of deterministic physical systems and is often produced due to the extreme sensitivity of the dynamical evolution of such a system to small changes in the initial conditions (11). A chaotic system abides by deterministic rules, but its evolution appears to be random, manifesting as non-linear dynamical systems that exhibit unpredictable behaviour (12). In other words, a chaotic system exhibits a stochastic behaviour within a deterministic system, making it challenging to predict its behaviour due to its high sensitivity to initial conditions.

4.2. Lyapunov Exponent.

Determining the presence of chaos in a dynamical system is a significant challenge, which can be tackled by calculating the largest Lyapunov exponent. By utilising this approach, researchers can identify the presence of chaos and obtain insights into the system's behaviour over time. The Lyapunov exponent measures the exponential separation between initially close state-space trajectories and provides an indication of the degree of chaos in the system (13).

The sign of the computed Lyapunov exponent indicates the degree of sensitivity to the initial conditions and provides a qualitative description of a system's dynamics, by indicating whether the system is chaotic or not. A positive Lyapunov exponent indicates divergence and chaos, while a negative Lyapunov exponent demonstrates convergence. On the other hand, the magnitude of the Lyapunov exponent highlights the time scale at which the dynamics of the system become potentially unpredictable (14). For the purpose of this section, which is mainly to detect chaos within the system, it is sufficient to derive only the largest Lyapunov exponents, rather than the entire Lyapunov spectrum as illustrated in other papers (10).

4.3. Largest Lyapunov Exponent Algorithm.

For each epoch in the data set, the exact locations of the earthquakes were identified by determining the precise longitudinal and latitudinal values for the epicentre of the earthquake, as shown in Table 1. In this study, we assume that the epicentre of the earthquake occurs at the spatial patch of the discretised fault with the highest slip potency rate for each epoch. For the case of computing the largest Lyapunov exponent, and since the earthquake is a single observable, we reconstruct a 2-dimensional system with variables x and y, where x is defined as the longitude and y is defined as the latitude of the epicentre of the earthquake at each epoch.

With the longitudinal and latitudinal values for the epicentre of the earthquake at each epoch, the MATLAB function *phaseSpaceReconstruction* was applied to both column vectors of the longitudinal and latitudinal values to approximate the time delay (lag) of the data, which is required to construct the logarithmic divergence against expansion step plot in the subsequent steps of the algorithm.

Due to the unknown dynamics and a lack of knowledge of the differential equations governing the underlying system describing earthquakes, the MATLAB function *lyapunovExponent* was utilised to compute the largest Lyapunov exponents purely from the data set alone, instead of manual computation.

The algorithm initializes by generating a delayed reconstruction $Y_{1:N}$ with embedding dimension of 2 and an approximated time delay of 10, as determined by the *phaseSpaceReconstruction* function earlier. For a point i, the algorithm identifies the nearest neighbour point i^* that satisfies the condition as follows:

$$\min_{i^*} ||Y_i - Y_{i^*}|| \qquad [11]$$

The condition must be satisfied such that $|i - i^*| > MinSep$, where $MinSep$ also known as the mean period of the data set, refers to the reciprocal of the mean frequency computed. Subsequently, the Lyapunov exponent for the entire expansion range is computed as follows (13):

$$\lambda(i) = \frac{1}{(K_{max} - K_{min} + 1)dt} \sum_{K=K_{min}}^{K_{max}} \frac{1}{K} \ln \frac{||Y_{i+K} - Y_{i^*+K}||}{||Y_i - Y_{i^*}||} \qquad [12]$$

K_{min} and K_{max} represents the minimum and maximum of the Expansion Range, which is utilised to approximate the local expansion rate for deriving the Lyapunov exponent. dt refers to the sampling time, which is equal to 1 day (86400 seconds) since the sampling rate is 1 data point per day. A single value for the largest Lyapunov exponent is then computed by deriving a best fit for a specific set of data points indicated by the expansion range to a polynomial as follows:

$$lyapExp = polyfit([K_{min}, K_{max}], \lambda(i)) \qquad [13]$$

4.4. Implications of a Chaotic System.

Figures 4a and 4b illustrate the logarithmic divergence against the expansion step plots, as well as the largest Lyapunov exponents, for both the longitudinal (x-data) and latitudinal values (y-data) of the earthquake epicentres.

From Figures 4a and 4b, it is evident that the largest Lyapunov exponents for both x-data and y-data are positive, which indicate the presence of chaos within the system. However, both exponents have small magnitudes which is mainly due to the small sampling frequency $fs = 1.157 \times 10^{-5}$ Hz, as only 1 data point is collected per day. The small magnitudes of the exponents highlight the slow rate of divergence of the infinitesimally close trajectories of the orbits of the earthquake epicentre in the phase space. Therefore, it can be concluded that in the long term, past a certain number of epochs, the system becomes increasingly difficult to predict, regardless of whether the underlying system is deterministic or not.

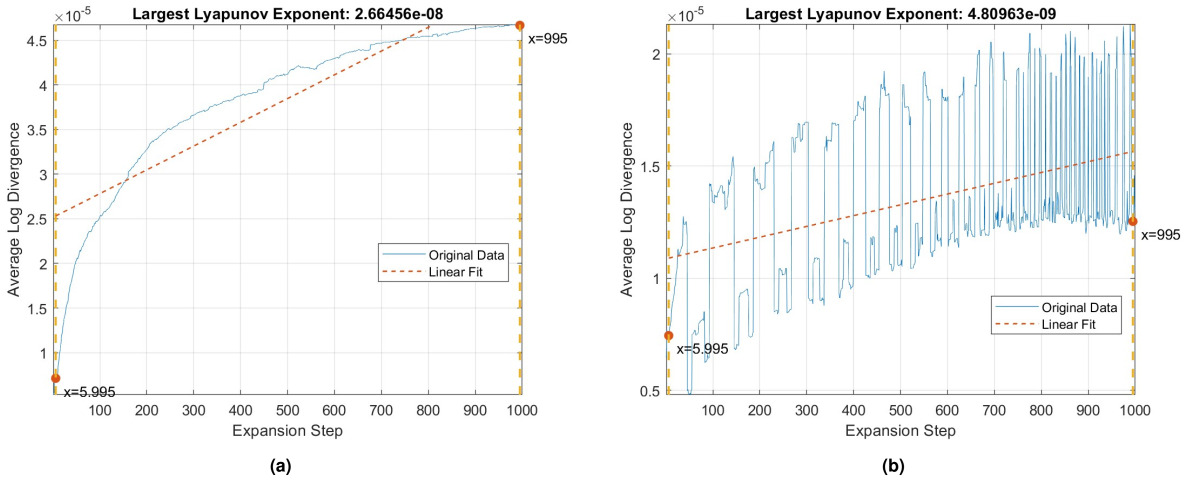


Fig. 4. Largest Lyapunov Exponent. An expansion range of 1000 is used for derivation of both plots. **(a)** Average logarithmic divergence against expansion step plot for x-data. **(b)** Average logarithmic divergence against expansion step plot for y-data.

5. Future state prediction of system

5.1. Complexity of future state prediction.

Future state prediction of dynamical systems is the ability to forecast the evolution of a system over time, based on its current state and the laws of physics or other relevant principles. A dynamical system can be condensed into a set of mathematical equations that describe how the state of the system changes over time. In order to make predictions about the future state of a dynamical system, it is necessary to have a sufficient prior knowledge of its underlying principles and dynamics, as well as accurate data on its current state. However, in the case of our spatial-temporal data set consisting of only slip potency rates at discretised plots of land, there is a lack of a clear governing model or set of equations to explain the dynamics of the system. Therefore, the future state prediction of the system discussed in this section are purely based on data-driven methods, such as DMD and neural networks.

5.2. Prediction using DMD.

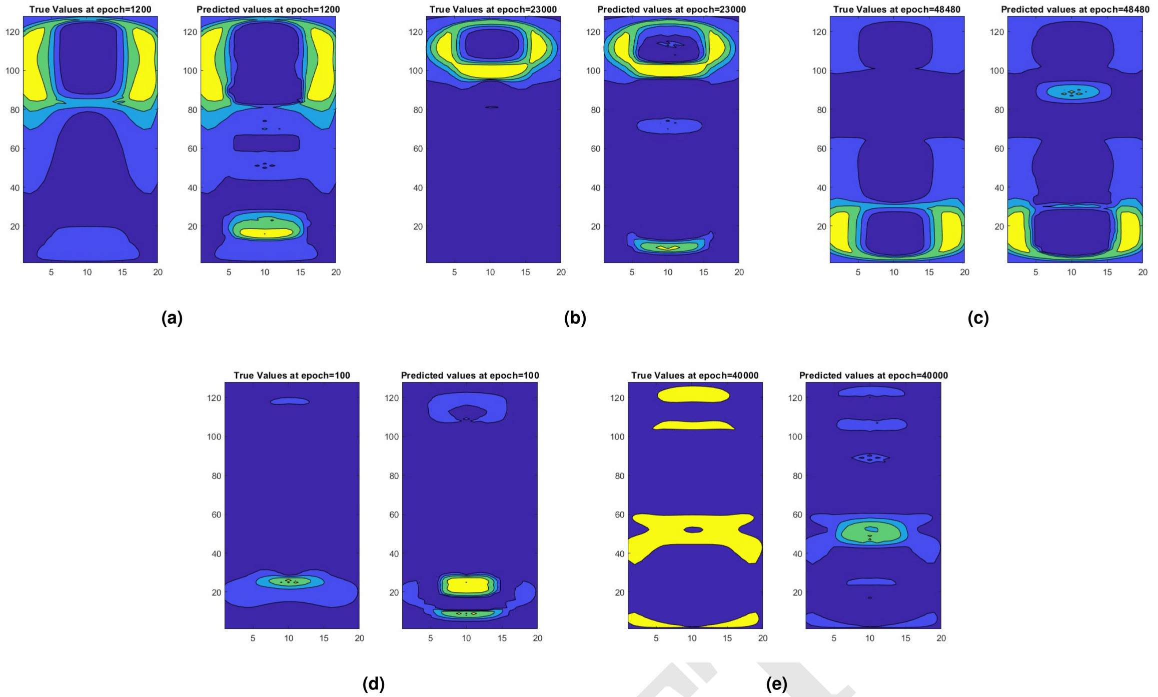
Before performing DMD on the data, it is important to remove any outliers or noise from the data to obtain accurate results. as the data is in the form of a 3D array, and there are several ways to filter it. In this case, Gaussian smoothing (15) is used to reduce the noise in the data. The smoothing is done using a Gaussian distribution with a standard deviation of $\mathbf{2}\sigma$. In MATLAB, the *imgaussfilt* function is used to perform this smoothing. The standard deviation determines the width of the kernel used in the smoothing process. A larger standard deviation will result in a wider kernel and stronger smoothing, while a smaller standard deviation will result in a narrower kernel and weaker smoothing. The smoothed data is utilised to generate the reduced-order model.

As explained earlier under the Dynamic Mode Decomposition sub-section, DMD aims to identify a best-fit linear operator $\mathbf{A}$, that provides a linear mapping from a state matrix $\mathbf{X}$ to another state matrix shifted by a time step, $\mathbf{X'}$. However, as the system is high-dimensional, the reduced-order model of the system is derived by computing its SVD and truncating the unimportant modes. Therefore, a total of 1000 modes is used to build the reduced-order model. Eventually, through DMD, the reduced matrix $\tilde{\mathbf{A}}$, which defines a linear model between state vectors shifted by a single time step, can be obtained.

After performing DMD on the spatial-temporal earthquake dataset, slip potency rate predictions were generated for various epochs. Figures 5a, 5b and 5c depict the comparison between the predicted slip potency rates and the true values at randomly chosen epochs 1200, 23000, and 48480 respectively. It can be observed that the predicted values show a similar trend to the true values but with some noise, upon comparison with the true values. The similarity between the predicted and true values suggests that the DMD algorithm was able to capture the underlying dynamics of the earthquake occurrences to some extent. However, the noise present in the predicted values indicates that the method may not have captured all the complex features of the dataset accurately.

On the other hand, from Figures 5d and 5e, the predicted slip potency rates at epochs 100 and 40000 showed irregularities with large noise when compared to the true values. This suggests that the DMD algorithm was not able to accurately capture the dynamics of the earthquake occurrences at these epochs. The irregularities and large noise in the predicted values may have resulted from the presence of outliers or other anomalous features in the dataset, which the DMD algorithm was unable to handle effectively.

The slip potency rate predictions obtained using the DMD algorithm provide valuable insights into the dynamics of earthquake occurrences, although the accuracy and effectiveness of this method can vary depending on the specific characteristics of the dataset being analysed. It is important to note that the DMD algorithm may not be suitable for non-linear systems, and



(a)

(b)

(c)

(d)

(e)

Fig. 5. Reduced order model predictions. (a,b,c) The predicted slip potency rates at epochs 1200, 23000, and 48480 show similarities with some noise when compared to the true values. **(d,e)** The slip potency rates predicted at epochs 100 and 40000 exhibit significant irregularities and noise when compared to the true values.

negative values may be produced if there is insufficient data between intermediate time points. For instance, if data is available only on a daily basis rather than hourly, the predictions may not be accurate. Thus, relying solely on DMD for predicting the behavior of a system may not be sufficient. Therefore, further investigations and improvements to the DMD algorithm may be necessary to enhance its accuracy and reliability in predicting slip potency rates and other related phenomena in earthquake occurrences.

5.3. Structure & Workings of Neural Networks.

Neural networks are a type of machine learning algorithm inspired by the fundamental structure and function of the human brain. Generally, they are used to learn relationships and patterns in data, and can be trained to make predictions or classifications based on input data. The driving purpose of neural networks is to handle complex problems that are difficult or impossible to solve using traditional programming techniques. Specifically, they are well-suited for problems that involve large amounts of data or complex relationships between variables, such as time-series forecasting.

The basic structure of a neural network consists of input nodes, hidden nodes, and output nodes. The input nodes receive input data, which is then passed through a series of weighted connections to the hidden nodes. The hidden nodes perform a non-linear transformation on the input data, which is then passed through another series of weighted connections to the output nodes. The output nodes generate the final output of the network, which can be used for prediction or classification tasks. The weights of the connections between the nodes are initially set to random values, and are then adjusted during the training process to minimize the error between the predicted output and the actual output (16). This process is usually executed based on an algorithm called back-propagation, which computes the gradient of the error with respect to the weights and subsequently adjusts the weights in the direction that minimizes the error.

Neural networks achieve their ability to model complex data through the use of non-linear activation functions (17) and multiple layers of hidden nodes. Non-linear activation functions, such as the *'sigmoid'* function and the hyperbolic tangent function *'tanh'*, enable the network to capture complex non-linear relationships between input and output variables, while multiple layers of hidden neurons allow the network to learn increasingly complex relationships and features in the data. Therefore, neural networks are a powerful and flexible tool for solving complex problems in a variety of domains, including earthquake prediction. Although they may require significant computational resources and may be generally difficult to interpret due to their 'black-box' nature, they have the potential to make useful and insightful predictions and understand complex systems.

5.4. Prediction using Neural Networks.

For our data set, we aim to design a neural network to derive a model that maps the state matrix $\mathbf{X}$ to another state matrix shifted by a time step, $\mathbf{X'}$. This is largely similar to the fundamental purpose of DMD, with the main difference being that

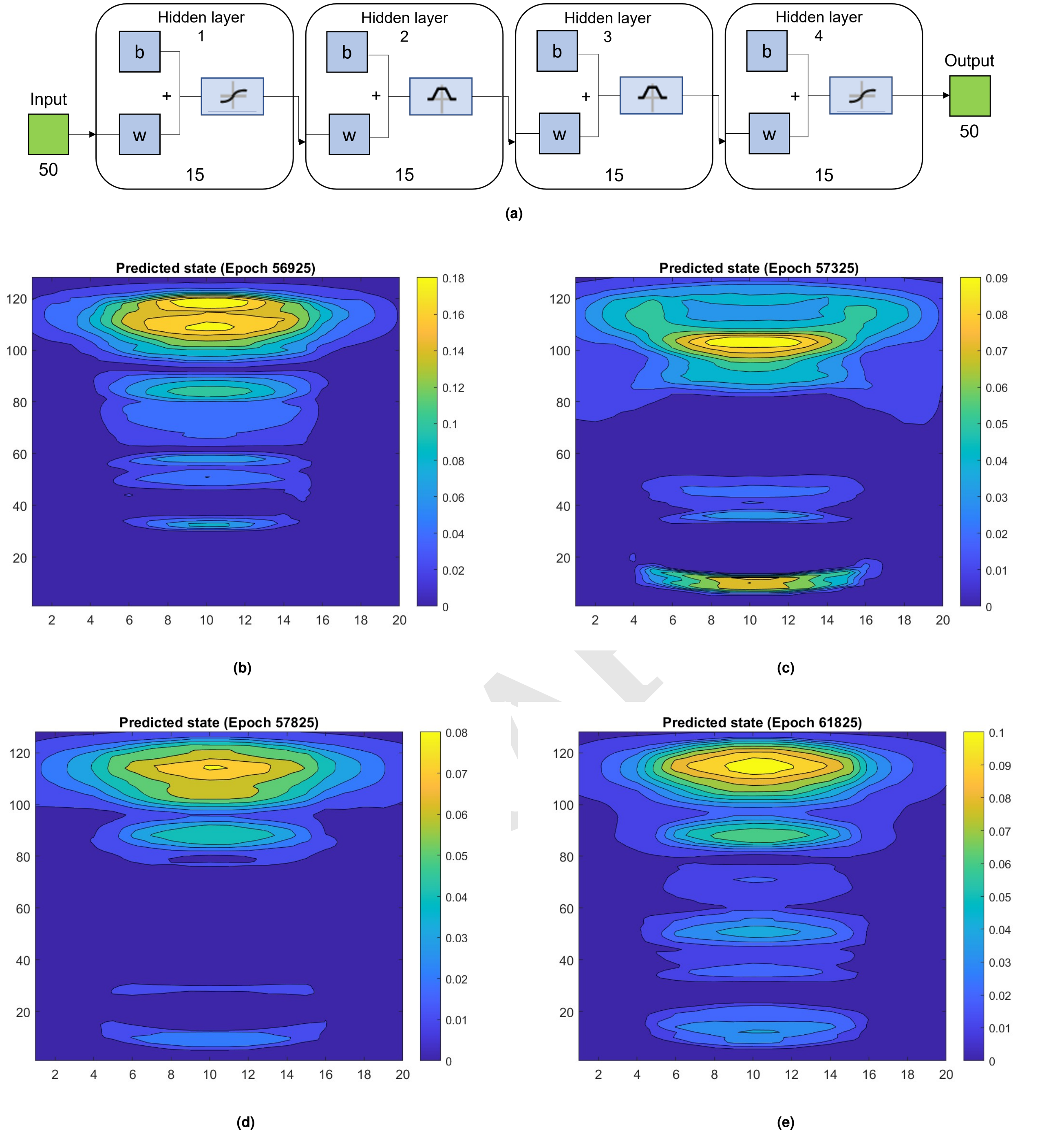


Fig. 6. **(a)** Neural network diagram used for future state prediction. **(b)** Future predicted state for epoch 56925, 100 epochs after the last given epoch in the data set. **(c)** Future predicted state for epoch 57325, 500 epochs after the last given epoch in the data set. **(d)** Future predicted state for epoch 57825, 1000 epochs after the last given epoch in the data set. **(e)** Future predicted state for epoch 61825, 5000 epochs after the last given epoch in the data set.

neural networks provide a non-linear mapping while DMD derives a best-fit linear mapping. Thus, neural networks should perform significantly better, especially if the data set is highly non-linear and complex.

Pre-processing of the data was performed, in order to reduce noise that is present, which results in possible reduction of computational time taken for training the neural network. Median filtering, which is a non-linear digital filtering technique, has been selected to pre-process the data due to its known effectiveness in noise reduction of a signal (18). In this study, a six-layer feed-forward Artificial Neural Network (ANN) is designed for predicting future states of our system. The algorithm supplies a sequential method for refining the weights in a back-propagation network to classify the provided input patterns accurately, given a training set of input–output pairs (19). The widely known gradient-descent method is utilised as the basis for the weight update algorithm (20). The six layers consist of one input layer, four hidden layers, and one output layer. To achieve a balance between computational time and accuracy of prediction, a total of 50 modes per epoch (truncated from the original 2560 modes) was fed as input into the neural network. The hidden layers each have 15 hidden neurons, with a tangent sigmoid (*'tansig'*) activation function implemented between the input layer and the first hidden layer, and also between the third and fourth hidden layers. A radial basis (*'radbas'*) transfer function was used between the first and second hidden layers, and also between the second and third hidden layers. A linear (*'purelin'*) transform function is chosen between the fourth hidden layer and the output layer. Due to its fast convergence for medium and large size neural networks (21), we selected the Levenberg–Marquardt (*'trainlm'*) algorithm is selected as the main back-propagation training method. To avoid overfitting of the network, 80% of the input data was used for training, 10% of the input data was used for validation, and the remaining 10% of the input data was used for testing. For visualization, the architecture of the neural network is illustrated in 6a.

Figures 6b, 6c, 6d, and 6e illustrate the predicted states at epochs 56925, 57325, 57825, and 61825 respectively. Through visualization, we see that the location enclosed by the region where the longitudinal values range from 100 to 120 and the latitudinal values range from 4 to 16. This is in tandem with the spatial structure of the system, as determined by the dominant DMD modes depicted in Figure 3b. Therefore, this indicates a broad characterisation of the system, whereby there is frequent clustering of energy in that specific enclosed spatial plot. However, as discovered in the Implications of a Chaotic System sub-section, it becomes challenging to predict the evolution of the system in the long-term due to the presence of chaos within the system. While neural networks are useful in providing insights of the system in the short-term, more tests and in-depth analysis have to be conducted to ascertain a longer-term prediction of the system dynamics.

6. Conclusion

In conclusion, this study has shed light on the complex dynamics of earthquake occurrences, using advanced data analysis techniques such as FFT, Gabor Transform, Chaos theory, DMD and Neural Networks. The results indicate a non-linear pattern, making it difficult to predict the system. Further investigations and improvements in the data analysis techniques may be necessary to improve the accuracy and reliability of predicting long-term behavior. For instance, the Sparse Identification of Non-Linear Dynamics (SINDy) algorithm (22) can be leveraged on to discover the governing equations of the dynamical system of earthquakes. The algorithm is designed to identify a relatively simple set of mathematical equations that accurately represent the underlying dynamics of the system. With more research and an increasing database of earthquake occurrences through the world, the difficulty of forecasting earthquakes could potentially be reduced, and bring about better measures in place to mitigate these dangerous phenomena.

ACKNOWLEDGMENTS. We would like to thank Dr. Gianmarco Mengaldo and Dr. Adriano Gualandi for their valuable feedback and guidance on the topics discussed in the manuscript. Finally, we would like to thank our families for their unwavering support throughout this research.

7. References

1. EO Brigham, R Morrow, The fast fourier transform. *IEEE spectrum* **4**, 63–70 (1967).
2. RN Bracewell, RN Bracewell, *The Fourier transform and its applications.* (McGraw-Hill New York) Vol. 31999, (1986).
3. AV Oppenheim, *Discrete-time signal processing.* (Pearson Education India), (1999).
4. MJ Bastiaans, Gabor's expansion and the zak transform for continuous-time and discrete-time signals in *Wavelet Analysis and Its Applications.* (Elsevier) Vol. 7, pp. 23–69 (1998).
5. G Rozza, M Hess, G Stabile, M Tezzele, F Ballarin, *1 Basic ideas and tools for projection-based model reduction of parametric partial differential equations*, eds. P Benner, et al. (De Gruyter, Berlin, Boston), pp. 1–47 (2021).
6. SL Brunton, JN Kutz, *7 Data-driven methods for reduced-order modeling*, eds. P Benner, et al. (De Gruyter, Berlin, Boston), pp. 307–344 (2021).
7. GH Golub, A Hoffman, GW Stewart, A generalization of the eckart-young-mirsky matrix approximation theorem. *Linear Algebr. its applications* **88**, 317–327 (1987).
8. T Krake, D Klötzl, B Eberhardt, D Weiskopf, Constrained dynamic mode decomposition. *IEEE Transactions on Vis. Comput. Graph.* **29**, 182–192 (2022).
9. G Tissot, L Cordier, N Benard, BR Noack, Dynamic mode decomposition of piv measurements for cylinder wake flow in turbulent regime in *Eighth International Symposium on Turbulence and Shear Flow Phenomena.* (Begel House Inc.), (2013).
10. FL Aderemi, OI Popoola, Computation of lyapunov exponent for characterizing the dynamics of earthquake. *Int. J. Recent Adv. Phys* **5**, 09–15 (2016).
11. F Heylighen, C Joslyn, RA Meyers, Encyclopedia of physical science and technology (2001).
12. H Goldstein, C Poole, J Safko, Classical mechanics (2002).
13. MT Rosenstein, JJ Collins, CJ De Luca, A practical method for calculating largest lyapunov exponents from small data sets. *Phys. D: Nonlinear Phenom.* **65**, 117–134 (1993).
14. A Wolf, JB Swift, HL Swinney, JA Vastano, Determining lyapunov exponents from a time series. *Phys. D: nonlinear phenomena* **16**, 285–317 (1985).
15. W Wang, J Gao, K Li, K Ma, X Zhang, Structure-oriented gaussian filter for seismic detail preserving smoothing in *2009 16th IEEE International Conference on Image Processing (ICIP).* pp. 601–604 (2009).
16. F Kulahci, A Özer, M Do? ru, Prediction of the radioactivity in hazar lake (sivrice, turkey) by artificial neural networks. *J. Radioanal. Nucl. Chem.* **269**, 63–68 (2006).
17. LV Fausett, *Fundamentals of neural networks: architectures, algorithms and applications.* (Pearson Education India), (2006).
18. S Tyan, Median filtering: Deterministic properties. *Two-Dimensional Digit. Signal Prcessing II: Transform. Median Filters* pp. 197–217 (2006).
19. F Külahcı, M İnceöz, M Doğru, E Aksoy, O Baykara, Artificial neural network model for earthquake prediction with radon monitoring. *Appl. Radiat. Isot.* **67**, 212–219 (2009).
20. D Graupe, *Principles of artificial neural networks: basic designs to deep learning.* (World Scientific), (2019).
21. J Zurada, *Introduction to artificial neural systems.* (West Publishing Co.), (1992).
22. SL Brunton, JL Proctor, JN Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. national academy sciences* **113**, 3932–3937 (2016).

8. Appendix

8.1. Derivation of FFT plots and Gabor Transform Spectrogram.

```
%% Derivation of FFT plots and Gabor Transform Spectrogram.
clc; clear; close all
```

```
% dimensions of data
n_samples = 56825;
n_latitudes = 20;
n_longitudes = 128;

% load data
ncfile = 'DalZilioetal2020sim.nc';

% visualize dataset content
ncinfo(ncfile)
ncdisp(ncfile)

% get SPR values
x = ncread(ncfile, 'SPR');

% get time snapshots
t = ncread(ncfile, 'time');

% get longitude values
lon = ncread(ncfile, 'lon');

% get latitude values
lat = ncread(ncfile, 'lat');

% Get the longitudes and latitudes of the maximum slip potency rates.
spr_mat = []; % Longitude, Latitude, Maximum SPR.
for i = 1:n_samples+1
    data = x(1:end, 1:end, i);
    max_spr = max(data(:));
    [ii,jj] = find(data == max_spr); % Row and column number.
    spr_mat(i,1) = ii;
    spr_mat(i,2) = jj;
    spr_mat(i,3) = max_spr;
end

spr_max = spr_mat(:,3);
f_noisy = spr_max;
dt = 1;

%% Compute the Fast Fourier Transform FFT
% Signal length.
n = length(t);

% Fast Fourier Transform.
fhat = fft(f_noisy,n);

% Power spectrum (power per each frequency).
PSD = fhat .* conj(fhat) / n;

% Create x-axis of frequencies in Hz.
freq = 1 / (dt*n) * (0:n);

% Plot only the first half of frequencies.
L = 1:floor(n/2);

%% Use the PSD to filter out noise within signal.
% Find all frequencies with large power above a certain threshold.
indices = PSD>125;

% Zero out all other frequencies.
PSD_clean = PSD .* indices;
```

```
% Zero out small Fourier coefficients.
fhat = indices .* fhat;

% Apply Inverse FFT for filtered time signal.
ffilt = ifft(fhat);

% Adjustments (Removing the sharp peak at 0Hz for FFT plot).
freq_adj = freq(1,2:end);
PSD_adj = PSD(2:end,1);
PSD_clean_adj = PSD_clean(2:end,1);

%% Plots for raw & clean signals, and FFT for raw & clean signals.
figure;
plot(t,f_noisy,'r','LineWidth',1), hold on
plot(t,ffilt,'b','LineWidth',1.5)
ylim([-0.25 1])
xlabel('Epoch (day)')
ylabel('Slip Potency Rate')
legend('Raw','Filtered', 'Location','southeast')
title('Plots for raw and filtered signals')

figure;
plot(freq_adj(L),PSD_adj(L),'r','LineWidth',1.5), hold on
plot(freq_adj(L),PSD_clean_adj(L),'-b','LineWidth',1.2)
xlim([0 0.005])
xlabel('Frequency (1/day)')
ylabel('Power Spectral Density')
legend('Raw','Filtered')
title('FFT plots for raw and filtered signals')

%% Spectrogram from Gabor Transform.
figure;
% window = 365 (1 year), noverlap = 292 (80% of window), nfft = 365.
% fs = 1/(24 * 3600) (sampling rate).
spectrogram(f_noisy,365,292,365,1/(24*3600),'yaxis');
title('Spectrogram of raw signal')


%% Coherent spatial structure and Ritz eigenvalues from DMD.
clc; clear; close all

% dimensions of data
n_samples = 56825;
n_latitudes = 20;
n_longitudes = 128;

% load data
ncfile = 'DalZilioetal2020sim.nc';

% visualize dataset content
ncinfo(ncfile)
ncdisp(ncfile)

% get SPR values
x = ncread(ncfile, 'SPR');

% get time snapshots
t = ncread(ncfile, 'time');

% get longitude values
lon = ncread(ncfile, 'lon');
```

```
% get latitude values
lat = ncread(ncfile, 'lat');

% Define the window size for the moving average filter.
% window_size = 10;

% Define filtering neighbourhood m x n.
m = 5;
n = 5;

% Using median filtering to smoothen data for all time epochs.
for i = 1:n_samples
    smooth_data(:,:,i) = medfilt2(x(:,:,i), [m n]);
end

% Flattened matrix.
X0 = reshape(smooth_data,128*20,[]);
[r,c] = size(X0);
X1 = X0(:, 1:c-1);
X2 = X0(:, 2:c);
% X1 = X0(:, 1:c-100);
% X2 = X0(:, 101:c);

% SVD
[U,S,V] = svd(X1,'econ');

% View plot to see which modes are dominant. (10 modes are dominant)
figure(1)
plot(diag(S)/sum(diag(S)),'ko','Linewidth',1.2)
hold on
plot(diag(S(1:200,1:200))/sum(diag(S)),'ro','Linewidth',1.2)
hold off
xlabel('Number of modes');
ylabel('Energy');
title('Energy vs Number of modes');

% Compute DMD (Phi are eigenvectors)
r = 200;  % Truncate at 200 modes.
U = U(:,1:r);
S = S(1:r,1:r);
V = V(:,1:r);
Atilde = U'*X2*V*inv(S); % Best-fit linear model.
[W,eigs] = eig(Atilde);
Phi = X2*V*inv(S)*W; % Reconstruct high-dim full state DMD e-vectors.
alpha1 = S * V(1,:)';
b = (W * eigs) \ alpha1;

%% Plot the dominant DMD modes.
num_modes = 10; % Choose the number of dominant modes to visualize.

real_dominant_modes = [];
imag_dominant_modes = [];

% Discard negative slip potency rates.
for i = 1:num_modes
    for j = 1:2560
        real_dominant_modes(j,i) = max([0 real(Phi(j,i))]);
        imag_dominant_modes(j,i) = max([0 imag(Phi(j,i))]);
    end
end
```

```
for i=1:num_modes
    figure;
    data_1 = reshape(real_dominant_modes(:,i),128,20);
    surf(data_1)
    xlabel('Latitude');
    ylabel('Longitude');
    zlabel('Relative Slip Potency Rate');
    title('Spatial Structure (Dominant DMD Mode)');
    colorbar
    figure;
    data_2 = reshape(imag_dominant_modes(:,i),128,20);
    surf(data_2)
    colorbar
end

%% Ritz Values.
figure
theta = (0:1:100)*2*pi/100;
plot(cos(theta),sin(theta),'r--') % plot unit circle
hold on, grid on
scatter(real(diag(eigs)),imag(diag(eigs)),'ok')
axis([-1.1 1.1 -1.1 1.1]);
title('DMD Spectrum (Eigenvalues/Ritz Values)');
xlabel('Re');
ylabel('Im');
```

8.3. Computation of Largest Lyapunov Exponents.

```
%% Computation of Largest Lyapunov Exponents
clc; clear; close all

% dimensions of data
n_samples = 56825;
n_latitudes = 20;
n_longitudes = 128;

% load data
ncfile = 'DalZilioetal2020sim.nc';

% visualize dataset content
ncinfo(ncfile)
ncdisp(ncfile)

% get SPR values
x = ncread(ncfile, 'SPR');

% get time snapshots
t = ncread(ncfile, 'time');

% get longitude values
lon = ncread(ncfile, 'lon');

% get latitude values
lat = ncread(ncfile, 'lat');

% Get the longitudes and latitudes of the maximum slip potency rates.
spr_mat = []; % Longitude, Latitude, Maximum SPR.
for i = 1:n_samples
    data = x(1:end, 1:end, i);
    max_spr = max(data(:));
    [ii,jj] = find(data == max_spr); % Row and column number.
    spr_mat(i,1) = ii;
    spr_mat(i,2) = jj;
    spr_mat(i,3) = max_spr;
end

% Using MATLAB lyapunovExponent function.
x_data = spr_mat(:,1); % Use value = 1 for x-data and value = 2 for y-data.
[~, elag, eDim] = phaseSpaceReconstruction(x_data); % Calculating lag and dimension.
fs = 1/(24*60*60); % Sampling frequency.
eRange = 1000;
lyapunovExponent(x_data, fs, elag, eDim, 'ExpansionRange', eRange)
```

8.4. Future state Prediction using DMD.

```
clc; clear; close all
%% this code is used to predict the future states using DMD
% dimensions of data
n_samples = 56825;
n_latitudes = 20;
n_longitudes = 128;

% load data
ncfile = 'DalZilioetal2020sim.nc';

% visualize dataset content
ncinfo(ncfile)
ncdisp(ncfile)

% get SPR values
x = ncread(ncfile, 'SPR');

% get time snapshots
t = ncread(ncfile, 'time');

% get longitude values
lon = ncread(ncfile, 'lon');

% get latitude values
lat = ncread(ncfile, 'lat');


%% Smoothening of original data

X_filt = x;
X_original = reshape(x,128*20,[]);
X_original = X_original(:,1:56826);
X_filt_flat = reshape(X_filt,128*20,[]);
X_filt_flat = X_filt_flat(:,1:56826);
[row,c] = size(X_filt_flat);
X1 = X_filt_flat(:, 1:c-1);
X2 = X_filt_flat(:, 2:c);
count = -1;

%% Plot to compare original with smoothened data
figure(1)
for i = [100,28000,56000]
    count = count+2;
    subplot(3,2,count)
    plot(X_original(:,i),'g')
    hold on

    plot(X1(:,i),'b')
    hold off
    xlim([0 2560])
    title(strcat("Flatened slip potenccy rate values at epoch=" , num2str(i)));
    legend('True values','filtered values');
end

%% Compute DMD
% SVD
[U,S,V] = svd(X1, 'econ');

r = 1000;  % Truncate at 10 modes. (Try 500 modes to see difference)
U = U(:,1:r);
S = S(1:r,1:r);
V = V(:,1:r);
Atilde = U'*X2*V*inv(S); % Best-fit linear model.
[W,eigs] = eig(Atilde);
Phi = X2*V*inv(S)*W; % Reconstruct high-dim full state DMD e-vectors.
alpha1 = S * V(1,:)';
b = (W * eigs) \ alpha1;

% Prediction of slip potency rates for epoch = 2.
alpha2 = Atilde * alpha1;
xt2 = U * alpha2;


%% Prediction of future state (epoch = 56826)

alpha = alpha1;
for i = 2:56826

    alpha_pre(:,i-1) = S*V(i-1,:)';
    alpha(:,i) = Atilde * alpha_pre(:,i-1);
    x_pre(:,i) = U * alpha(:,i);

end
x_pre = max(x_pre,0) ;
X_predected = reshape(x_pre,128,20,56826);


%% visualisation

figure(1)
count = 0;
for i = [100,28000,56000]
    count = count+2;
    subplot(3,2,count)

    plot(X1(:,i),'b')
    hold on

    plot(x_pre(:,i),'r')
    hold off

    xlim([0 2560])
    title(strcat("Predicted value vs Filtered values at epoch=" , num2str(i)));
    legend('Filtered values','Predicted values');
end
```

```
% figure(2)
levels = [0,0.01,0.03,0.05,0.08,20,100,660];
count = 0;
fig = 2;
for i = [1200,23000,48480,100,40000,50000]

    figure(fig)
    fig= fig +1;
    count = count+1;
    subplot(1,2,count)

    contourf(X_filt(:,:,i),levels)
    title(strcat("True Values at epoch=" , num2str(i)))
    count = count+1;
    subplot(1,2,count)
    contourf(X_predected(:,:,i),levels)
    title(strcat("Predicted values at epoch=" , num2str(i)));
%     legend('Filtered values','Predicted values');
%     xlim([0 2560])
    count =0;
    hold off
    pause(0.1)
end
```

8.5. Designing Neural Network for future state prediction.

```
%% Designing Neural Network for future state prediction.
clc; clear; close all

% dimensions of data
n_samples = 56825;
n_latitudes = 20;
n_longitudes = 128;

% load data
ncfile = 'DalZilioetal2020sim.nc';

% visualize dataset content
ncinfo(ncfile)
ncdisp(ncfile)

% get SPR values
x = ncread(ncfile, 'SPR');

% get time snapshots
t = ncread(ncfile, 'time');

% get longitude values
lon = ncread(ncfile, 'lon');

% get latitude values
lat = ncread(ncfile, 'lat');

%% Smoothening the data.
% Define filtering neighbourhood m x n.
m = 5;
n = 5;

% Using median filtering to smoothen data for all time epochs.
for i = 1:n_samples
```

```
    smooth_data(:,:,i) = medfilt2(x(:,:,i), [m n]);
end

future_time = 5000;
for k = 1:50:future_time
    X0 = reshape(smooth_data,128*20,[]);
    [r,c] = size(X0);
    X1 = X0(:, 1: c-k);
    X2 = X0(:, 1+k:c);

    % SVD
    [U,S,V] = svd(X1,'econ');

    % Compute DMD (Phi are eigenvectors)
    r = 50;  % Truncate at 10 modes. (Try 500 modes to see difference)
    U1 = U1(:,1:r);
    S1 = S1(1:r,1:r);
    V1 = V1(:,1:r);
    U2 = U2(:,1:r);
    S2 = S2(1:r,1:r);
    V2 = V2(:,1:r);
    alpha1 = S1 * V1(:,:)';
    alpha2 = S2 * V2(:,:)';
    x_norm = normalize(alpha1);

    % Neural network configuration and training
    net = feedforwardnet([15 15 15 15]);

    net.layers{1}.transferFcn = 'tansig';
    net.layers{2}.transferFcn = 'radbas';
    net.layers{3}.transferFcn = 'radbas';
    net.layers{4}.transferFcn = 'tansig';
    net.layers{5}.transferFcn = 'purelin';

    net.divideFcn = 'dividerand';
    net.divideMode = 'sample';
    net.divideParam.trainRatio = 0.8;
    net.divideParam.valRatio   = 0.1;
    net.divideParam.testRatio  = 0.1;

    net.trainFcn = 'trainlm';
    net.trainParam.max_epochs = 1000;
    net.trainParam.goal = 0.01;
    net = train(net, x_norm, alpha2);

    %% Removing negative slip potency rates.
    for i = 1:2560
        xt2(i,1) = max([0 xt2(i,1)]);
    end

    %% Plotting contour plots.
    % Predicted states (For epoch = last + 1000).
    contourf(reshape(xt2(1:2560,1),128,20))
    colorbar
    title(strcat("Predicted state (Epoch ", num2str(n_samples + k), ")"));
end
```