Data-Driven of Spatio-temporal Prediction: Earthquakes Case

Huang Yi[a], **Zhang Yumeng**[a], **and Luo Wenjun**[a]

[a]National University of Singapore

This manuscript was compiled on June 19, 2023

Spatio-temporal (ST) earthquakes prediction is significant for prevention of damages. In this report, there is a dynamic high-dimensional dataset of SPR sampling from two-dimensional space. This report proposes a flow which can be used in general dynamic ST system. The flow includes dataset preprocessing based on analysis, two techniques to reduce the order of this system, and four prediction techniques. First, after visulizing the dataset and analyzing the dataset in space, time, and frequency aspects, the outliers, which are regarded as noise, are removed. Meanwhile, through system eigenvalue asnd energy of modes, system predictability is judged. Then, in order to reduce system complexity, order reduction techniques are applied. One of the reduction techniques is based on the PCA technique. Another is dynamic mode decomposition (DMD) with Koopman which is widely applied in the dynamic system. Inversing of reduced order system is also important, this report compares the reduction techniques from accuracy of prediction and inversing accuracy. Finally, the main-stream black box time series models, including LSTM, RNN, and Transformer, are applied. Those black box models are compared with white box technique which is dynamic mode decomposition (DMD) with Koopman.

Spatio-temporal Prediction | Reduced-order model | Dynamic mode decomposition (DMD) |

Introduction. An earthquake is a seismic vibration of the Earth's surface caused by the movement of tectonic plates within the Earth's interior. Although various methods are available for predicting earthquakes, accurate earthquake prediction is still a difficult task due to the complexity and unpredictability of earthquakes, which are influenced by many factors. Historical earthquake data can provide useful information, and as such, historical data is often used to predict earthquakes.

Three NN (neural network) structures commonly used for time-series datasets are LSTM (Long Short-Term Memory), Transformer, and RNN (Recurrent Neural Network). LSTM is a variant of RNN that can handle long sequence data and has memory capability. Transformer uses self-attention to process sequence data, and it can perform parallel computation, making it faster than LSTM for handling long sequences. RNN is a basic neural network structure that remembers previous inputs for subsequent predictions. LSTM and RNN can be used to model time-series data and have performed well in earthquake prediction tasks. The Transformer model uses self-attention mechanisms to learn the relationships between different positions in a sequence, and it has also been used in earthquake prediction tasks in recent years.

Recently, data-driven methods such as dynamic mode decomposition (DMD) have shown promise in predicting the behavior of complex systems. DMD is a data analysis and prediction technique that decomposes complex dynamic systems into several simple dynamic modes. By analyzing these dynamic modes, the behavior of the system can be understood and future trends can be predicted. The basic idea of DMD is to decompose the dynamic behavior of the system into a set of basic modes, which can be obtained by singular value decomposition (SVD) of the observed data of the system. Therefore, DMD is a data-based method that does not require a deep understanding of the internal mechanisms of the system.

In this report, we discuss the applications of the above methods in earthquake prediction and compare the results obtained by different models. The main purpose of this study is to investigate the performance of DMD in earthquake prediction and compare it with existing methods.

Lecture Review. LSTM and RNN are commonly used deep learning models for processing sequential data. When dealing with time series data, it is necessary to choose an appropriate neural network structure and adjust it according to the size and complexity of the dataset. Li et al.[1] modeled the soil seismic response of KiK-net borehole array stations using Convolutional Neural Network (CNN) and Long Short-Term Memory Neural Network (LSTM) models and the results show that CNN and LSTM models can accurately predict soil seismic responses at different frequencies, which is of practical significance for earthquake engineering and earthquake disaster management. Laurenti et al.[2] used a deep neural network consisting of LSTM and CNN to train laboratory earthquake signals before the earthquake occurred to predict the time of earthquake occurrence, indicating that deep learning models have great potential in earthquake prediction and stress prediction, providing useful tools for earthquake and geology. Abebe et al.[3] used deep learning to predict the magnitude of earthquakes in the Horn of Africa region. The results show that using these two models can effectively predict the magnitude of earthquakes.

At the outset of the project, a discussion is held among team members to formulate and refine the project, and allocate tasks based on the estimated workload. Luo Wenjun assumes responsibility for overseeing system analysis, PCA model reduction, Transformer algorithm prediction, and producing the corresponding sections of the report. Huang Yi is tasked with leading the prediction using LSTM and RNN algorithms, conducting a comparative analysis of the three algorithms, and authoring the relevant report section. Zhang Yumeng undertakes order reduction and prediction using DMD, conducts a comparative analysis of DMD against other prediction methods, and finished the corresponding report sections.

Transformer is a neural network architecture based on self-attention mechanism used for processing sequential data. Franco et al.[4] proposed a novel trajectory prediction method based on the Transformer model and evaluated it experimentally, providing important references and insights for our in-depth understanding and application of Transformer models for trajectory prediction. Zhang et al.[5] proposed an earthquake prediction method based on the Transformer model, called EPT. EPT trained a data-driven Transformer model using earthquake data to predict the time, location, and magnitude of earthquakes. The experimental results show that EPT has high accuracy and stability in predicting earthquake time, location, and magnitude, providing important references and insights for our in-depth understanding and application of earthquake prediction methods based on Transformer models. Li et al.[6] compared the performance of different machine learning models, including GBDT, SVM, ANN, and Transformer, in predicting explosion loads. The authors used a real explosion dataset to evaluate the performance of these models and analyzed their advantages and disadvantages. This results provides important references and insights for selecting suitable machine learning models for prediction. Jiang et al.[7] compared the performance of two earthquake detection methods, PhaseNet and EQTransformer, in detecting two real earthquake events (Yangbi and Maduo). The experimental results show that EQTransformer has higher accuracy and robustness in detecting earthquake events, especially in handling low signal-to-noise ratio data. However, PhaseNet performs better in handling high-frequency events. Finally, the authors discussed the application scenarios and future research directions of the two methods. This paper provides useful information and references for selecting suitable earthquake detection methods.

PCA (Principal Component Analysis) is a statistical method used for data dimensionality reduction and feature extraction. Although it cannot be directly used for earthquake prediction, it can be effectively applied to earthquake data processing. PCA can be used to reduce the dimensions of multidimensional data, decrease redundant information, and improve data visualization and interpretability. For instance, PCA can be applied to analyze multiple parameters of earthquakes, such as magnitude, source depth, and epicenter location, and transform them into fewer and more representative principal components, which can help better understand earthquake data. Chang et al.[8] proposed a technique called "sliding PCA" method, which can detect ionospheric anomalies before earthquakes. This method uses PCA to extract the variation patterns of Total Electron Content (TEC) in the ionosphere, establishes a baseline model by analyzing historical data, and then compares new TEC data to detect any abnormal changes. Lin[9] proposed a method based on Nonlinear Principal Component Analysis (NLPCA) to study ionospheric anomalies before the Wenchuan earthquake in China. The author used data from Ground-based Global Navigation Satellite Systems (GNSS) receivers to measure TEC and analyzed the nonlinear features of TEC data using NLPCA. This study provides an analysis tool based on NLPCA method that can be used to investigate ionospheric anomalies before earthquakes.

DMD (Dynamic Mode Decomposition) is a technique for analyzing and predicting the behavior of dynamic systems. It decomposes a complex system into a set of basic dynamic modes using singular value decomposition (SVD), which can predict the system's future behavior. DMD is useful in real-time control and monitoring applications, and its accuracy depends on data quality and system complexity. Koopman is a mathematical theory that describes the evolution of dynamic systems based on the Koopman operator, which can describe the linear behavior of nonlinear systems. Koopman and DMD are often used in combination to analyze dynamic systems. Lu et al.[10] evaluated the prediction accuracy of Dynamic Mode Decomposition (DMD) as a data-driven control (DDC) method. The results show that DMD can achieve high prediction accuracy in some cases, but prediction errors may occur in other cases. Williams et al.[11] introduced how to extend data-driven Koopman analysis to systems with control inputs. This method can be used for controller design, system behavior prediction, stability and fault diagnosis of the system. Hirsh et al.[12]proposed a data-driven spatio-temporal modal decomposition method based on a data-driven time-frequency analysis. This method can extract a set of basic functions from spatio-temporal data to achieve dimensionality reduction and denoising processing of the data, as well as extract frequency information that changes over time and space. Katrutsa et al.[13] proposed an optimal prediction method based on the t-model for extending the Dynamic Mode Decomposition (DMD) method to dynamic systems with incomplete information.This method can be used for reconstructing dynamic systems from incomplete data, predicting their future behavior, and studying the stability and control of the system. Brunton et al.[14] introduced a method for extracting spatio-temporal coherent patterns in large-scale neural recordings using the Dynamic Mode Decomposition (DMD) method. Through simulations and actual data analysis of large-scale neural recordings, the authors demonstrated the effectiveness and practicality of this method and its potential applications in understanding the spatiotemporal dynamics of the neural system and analyzing neurological diseases. Abu-Seif et al.[15] proposed a data-driven modeling method for lithium-ion batteries using the Dynamic Mode Decomposition (DMD) method. This method can be used for designing battery control algorithms, predicting battery life and performance, and optimizing battery design and manufacturing.

In summary, DMD is suitable for processing time-series data and has better processing effects for nonlinear and non-stationary signals. It can be used for earthquake prediction, but requires more computational resources and more data to obtain accurate results. When using the DMD method to predict earthquakes, data collected before the earthquake must be used to build the model. Specifically, data including seismic activity before the earthquake, geophysical and geological changes before the earthquake, and environmental changes before the earthquake need to be collected and preprocessed. Next, the data is decomposed and modeled using the DMD method. The DMD method can decompose time-series data into a set of modes, each representing a vibration pattern. By analyzing the relationship between these modes, a prediction model can be established and used to predict future earthquakes. It should be noted that earthquake prediction is a very difficult task, and relying solely on a single method cannot achieve good results. When using the DMD method to predict earthquakes, it is best to combine it with other methods for analysis and prediction to improve the accuracy of the prediction.

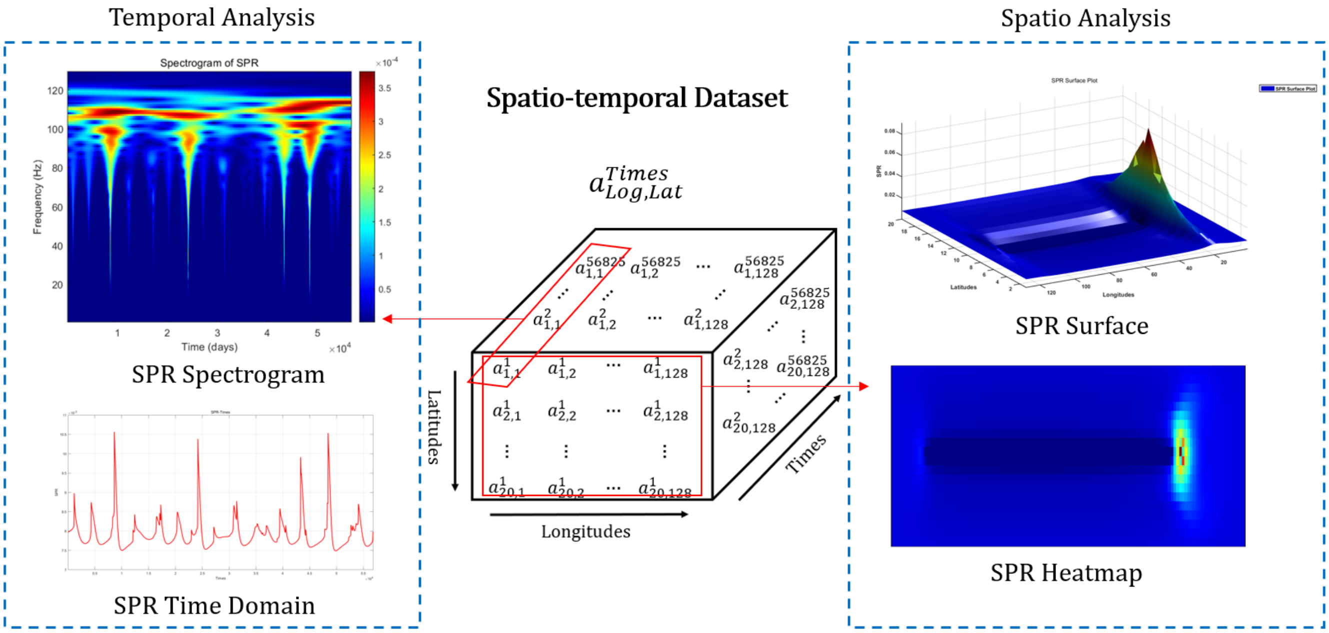


Fig. 1. Spatio-Temporal Dataset Analysis

Database Analysis and Pre-processing. Building accurate earthquake prediction models requires the use of large spatio-temporal datasets that capture the complex behavior of seismic systems. In this context, researchers use data-driven machine learning and time series prediction techniques to analyze such datasets and forecast future seismic events. To illustrate the data required for such analysis, let's consider a spatio-temporal dataset that mimics complex earthquake occurrences on a specified planar fault. The fault has been discretized into 20 x 128 patches, with each patch measuring 2.5 x 2.5 km2. The dataset contains a total of 56,825 time snapshots. Such a dataset provides a wealth of information that can be used to train machine learning models and make accurate predictions about future earthquakes. By analyzing this data, researchers can identify patterns and trends in the seismic activity that may provide insight into the behavior of the underlying fault system. The Fig. 1 contains a spatial and temporal analysis of the data set.

Temporal Analysis. Temporal analysis refers to the process of analyzing data over time to identify patterns and trends. Time series analysis involves modeling data over time to identify patterns, trends, and seasonality. Trend analysis focuses on identifying long-term trends in data and can be used to forecast future values.

The dataset consists of time series data from 2560 patches, and we select one patch for analysis. The patch contains 56825 data points, which are collected over time and arranged in a (56825,1) two-dimensional matrix. Fig.1 shows the data which is visualized over time, revealing a noticeable degree of periodicity.

The fundamental principle of wavelet transform is to decompose a signal into a sequence of wavelets of varying scales and translation positions, extracting different frequency components of the signal and obtaining the signal's time-frequency distribution. In the resulting wavelet coefficients, high-frequency wavelet coefficients represent instantaneous changes in the signal, while low-frequency wavelet coefficients represent slow changes in the signal. Therefore, wavelet transform can provide more refined and rich signal information in both the time and frequency domains, and it can be applied in areas such as signal compression, noise filtering, and signal feature extraction.

$$c_{j,k} = \langle a, \psi_{j,k} \rangle = \frac{1}{\sqrt{2^j}} \sum_{n=0}^{N-1} a[n] \overline{\psi\left(\frac{n-2^j k}{2^j}\right)} \qquad [1]$$

Based on the discrete wavelet transform, time-domain and frequency-domain analyses can be performed on the time series of one of the patches. Assuming the data is represented as $a[n]$ and the wavelet basis function is $\psi_{j,k}(n)$, the discrete wavelet transform coefficients of the data can be represented by equation [1]. Here, j and k represent the scale and translation values of the discrete wavelet transform, respectively, and N is the length of the time series data. Using this equation, the original time series can be decomposed into multiple wavelet coefficients, thereby enabling analysis and processing in both the time and frequency domains. From the SPR spectrogram in fig. 1, the majority of the energy is concentrated in the frequency range of 80-120Hz, which corresponds to the peak value in the time domain.

Spatial Analysis. Spatial analysis involves visualizing the SPR values at different latitude and longitude coordinates. The spatial analysis shown in Figure 1 visualizes the SPR values measured in all the patches at the first time point. In the SPR surface plot, the x- and y-axis coordinates represent longitude and latitude, respectively, with a total of 128 and 20 sampling points, respectively. Additionally, the spatial data is represented as a heatmap. These two plots indicate that high SPR values are concentrated in a specific area during this time period, with a noticeable drop in SPR values in other areas. In terms of predictability, it can be seen from the spatial analysis that the patches are continuous in space, and therefore, some areas can be reduced in dimensionality to reduce the computational complexity of prediction.

Principal Component Analysis. Given the high-dimensional of the dataset with dimensions (56825, 20, 128), overfitting, lower interpretability, and increased computational complexity are possible, thus necessitating the reduction of the model's order. One popular approach is to use Principal Component Analysis (PCA), which involves identifying the linear combinations of features that capture the most variance in the data and projecting the data onto a lower-dimensional subspace defined by these combinations. [16]

Outlier Removal. Outliers can have a disproportionate impact on the results of PCA, so it is generally recommended to remove outliers before performing PCA. Due to the PCA technology aims to identify the most important features in a dataset by finding the directions of maximum variance in the data, outliers can introduce a significant amount of variance into the dataset, which can distort the principal components identified by PCA. This can lead to inaccurate results, as the principal components may not be representative of the underlying structure of the data.

Additionally, outliers can affect the covariance matrix used in PCA, which is calculated from the mean-centered data. If there are outliers in the data, the mean may not be a representative measure of central tendency, and the covariance matrix may not accurately reflect the relationships between the features in the dataset.

However, it is important to note that removing outliers can also affect the results of PCA, as it can lead to a loss of information and potentially bias the results. Therefore, it is important to use a robust method for outlier detection and removal and to carefully consider the impact of outlier removal on the analysis results. As the data distribution exhibits a high degree of concentration and periodicity, the determination of the outlier removal threshold is relatively straightforward. Specifically, visual inspection is employed to select a threshold that minimizes the impact on peak values while ensuring that outlier removal is effective. In this particular case, a threshold of 1.5 was found to be appropriate. The Fig. X. illustrates the performance of with and without outliers.

Model Order Reduction. To apply PCA to this data set, it must first be reshaped into a 2D matrix by flattening the original 3D data array, resulting in a (56825, 2560) matrix. The mathematical expression for the PCA algorithm is provided below.

Step 1. Identify the principal components of the data set, which are the most representative feature vectors:

$$X_c = X - \frac{1}{56825}\sum_{i=1}^{56825} x_i \qquad [2]$$

Step 2. Calculate the covariance matrix of the centralized data set:

$$C = \frac{1}{56825 - 1} X_c^T X_c \qquad [3]$$

Step 3. Compute the eigenvectors and eigenvalues of C:

$$C = U\Sigma U^T \qquad [4]$$

Step 4. Sort the eigenvectors based on their corresponding eigenvalues:

$$U_{sort} = [u_1, u_2, ..., u_{2560}] \qquad [5]$$

Step 5. Select the top 20 eigenvectors to form a new 20-dimensional feature space:

$$U_{20} = [u_1, u_2, ..., u_{20}] \qquad [6]$$

Step 6. Project the original data onto the new 20-dimensional feature space:

$$Y = X_c U_2 0, Y = Y[:, : 20] \qquad [7]$$

Upon reducing the order of the model, it is imperative to conduct a rigorous validation process to ascertain the accuracy of the reduced model in capturing the fundamental behavior of the original model. The validation process entails a meticulous comparison between the anticipated output of the reduced model and the actual output of the original model, which is performed over an extensive range of inputs. It is noteworthy to emphasize that this critical stage of model reduction and validation serves as a crucial step in ensuring the reliability and efficiency of the reduced model in effectively representing the dynamics of the original model. Therefore, the meticulousness and comprehensiveness of the validation process are paramount in guaranteeing the successful implementation and application of the reduced model in various relevant contexts.[17, 18] In this case, the predictive algorithms employed comprise PyTorch's nn transformer, as well as the Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) models.

PyTorch's nn Transformer. This chapter outlines an approach for the prediction of time series using PyTorch's nnTransformer() module in combination with Principal Component Analysis (PCA). It can be used to verify whether the low-dimensional model obtained using PCA effectively represents the original model and achieves good enough predictions. The process of implementing this approach is as follows, with each step building upon the previous ones to optimize the model's prediction performance:

Step 1. Sliding Window Encoding: Encode the data after reducing order by PCA into a set of sliding windows. Each consists of a fixed number of time steps, which is 10 in this case. These sliding windows are then used as input to the Transformer model, which will be designed in the next step.

Step 2. Transformer Architecture: Design the Transformer architecture by specifying the number of layers, the number of heads and the size of the hidden layers, which is used to train a Transformer model on the sliding window-encoded data to predict the value of the next time step.

Step 3. Model Training: Train the Transformer model on sliding window-encoded data using appropriate optimization algorithms.

Step 4. Model Validation: Validate the performance of the Transformer model using appropriate metrics. The loss function used to train the model is typically Mean Squared Error (MSE) or a variant of it, which measures the difference between the predicted and actual values of the time series.

Step 5. Model Tuning: Fine-tune the hyperparameters of the Transformer model to optimize its performance. The optimization algorithm used to train the model is usually Adam or a variant of it, which performs stochastic gradient descent with adaptive learning rates.

Step 6. Prediction: Use the trained Transformer model to make predictions on data that has undergone the PCA and compare with the true value.

Based on the aforementioned process, the nn Transformer offers two noteworthy advantages. Firstly, it boasts a high degree of flexibility, as it is highly customizable and allows for the tuning of various hyperparameters such as the number of layers, number of heads, and learning rates to optimize performance on a given dataset. After careful testing and analysis, it was determined that the optimal number of layers and heads for our purposes were 1 and 4, respectively. Additionally, the adaptability of the nn Transformer is a feature worthy of recognition. Through the utilization of sliding window encoding and attention mechanisms, the Transformer model is capable of accommodating various types of time series data, including those with irregular time intervals or missing values.[19]

Therefore, this approach allows for the efficient capture of temporal dependencies in the data, resulting in improved accuracy in time series forecasting. The results will be compared and discussed with the other two methods in the Chapter named Comparison of Transformer, RNN and LSTM.

Recurrent Neural Network. Feeding the data by reducing the order using PCA to a Recurrent Neural Network (RNN) for prediction is a common technique in machine learning.

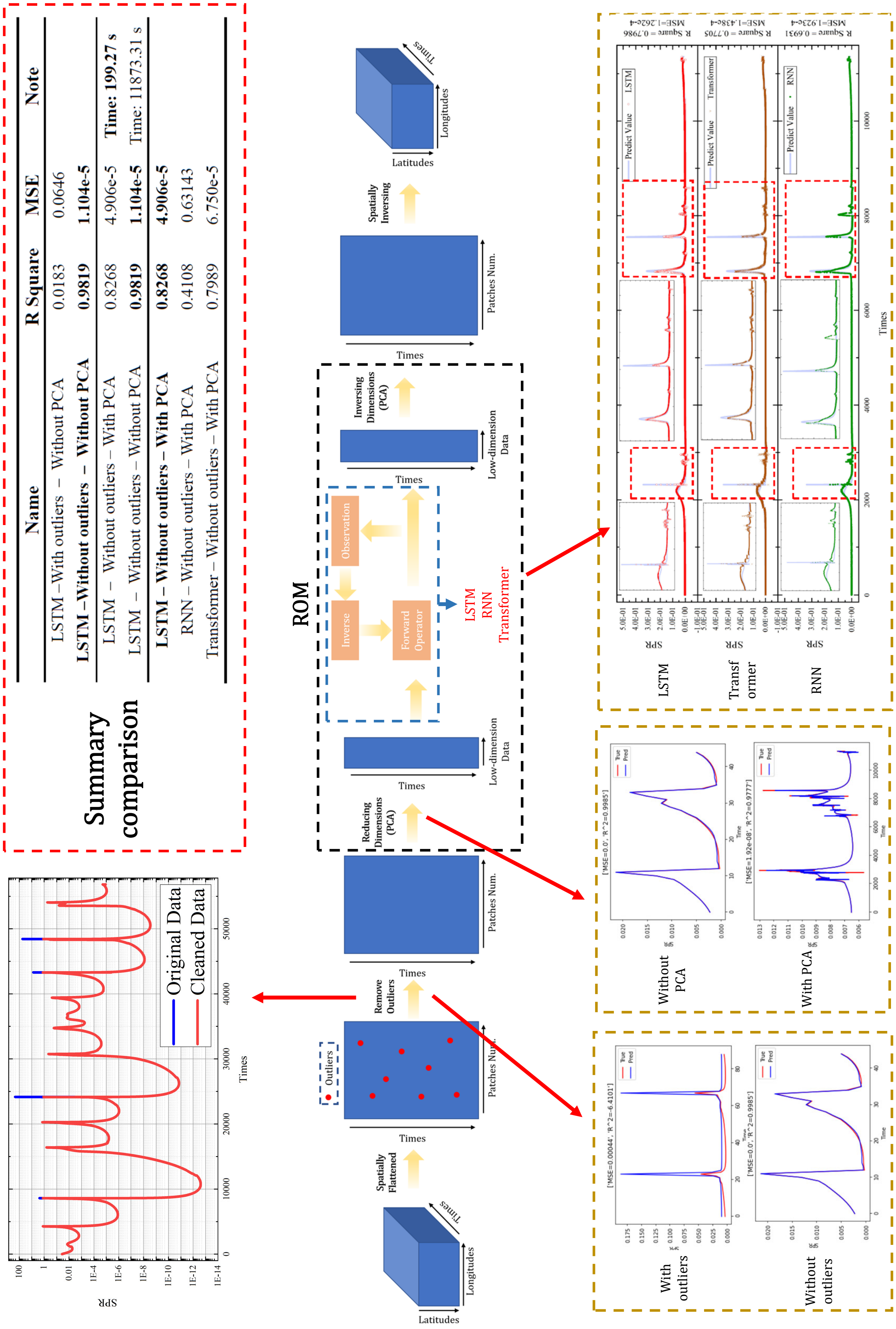
The first step is to apply PCA to the input data to obtain a set of principal components that capture the most important features of the data. These principal components can then be used as inputs to the RNN for prediction. The number of principal components to retain is a hyperparameter that can be tuned to optimize model performance. Once the principal components have been obtained, they can be fed to the RNN as input sequences. The RNN will then learn to predict the next value in the sequence based on the previous values. The output of the RNN can be mapped back to the original feature space using the inverse PCA transform to obtain the final prediction.[20] The results will be compared and discussed with the other two methods in the Chapter named Comparison of Transformer, RNN and LSTM.

Long Short-Term Memory. Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that excels in learning long-term dependencies in sequential data. Unlike traditional RNNs, which suffer from the vanishing gradient problem and have difficulty learning such dependencies, LSTMs are designed with memory cells that are connected to input, output, and forget gates. The input gate controls how much of the new input should be stored in the memory cell, while the forget gate determines how much of the previous cell state should be discarded. The output gate controls how much of the current cell state should be used as the network's prediction.[20]

In addition to their ability to maintain information over long periods of time, LSTMs have several other advantages. For instance, they are robust to noisy or missing data, as they can selectively retain and discard information in the memory cell. Furthermore, LSTMs are capable of modeling non-linear relationships in the data, enabling them to capture complex patterns and make accurate predictions.

Given these advantages, the current project has chosen to test the performance of LSTM in model prediction, even though traditional RNNs have already been used. The results will be compared and discussed with the other two methods in the Chapter named Comparison of Transformer, RNN and LSTM.

Comparison of Transformer, RNN and LSTM. Comparing various prediction methods can be a useful approach to assessing the accuracy and effectiveness of order reduction using PCA. Fig. 2 provides a comparison and summary of the three forecasting methods discussed earlier. The figure also highlights the distinctions between outlier separation and model reduction, both of which are based on LSTM.



Summary comparison

Name	R Square	MSE	Note
LSTM –With outliers – Without PCA	0.0183	0.0646	
LSTM –Without outliers – Without PCA	**0.9819**	**1.104e-5**	
LSTM – Without outliers – With PCA	0.8268	4.906e-5	**Time: 199.27 s**
LSTM – Without outliers – Without PCA	**0.9819**	**1.104e-5**	Time: 11873.31 s
LSTM – Without outliers – With PCA	**0.8268**	**4.906e-5**	
RNN – Without outliers – With PCA	0.4108	0.63143	
Transformer – Without outliers – With PCA	0.7989	6.750e-5	

Fig. 2. Summary of the Prediction based on Reduced-order Model

Two important performance indicators for evaluating the quality of predictions are the coefficient of determination and mean squared error. The coefficient of determination, also known as "R-squared (R^2)", is a common metric used to assess the accuracy of regression models in predicting continuous outcomes. It represents the proportion of the dependent variable's variance explained by the independent variables in the model. A perfect fit is indicated by a value of R^2 of 1, meaning that all the variance in the dependent variable can be explained by the independent variables, while a value of R^2 of 0 indicates no linear relationship between the variables. Generally, higher R^2 values indicate better fit and greater predictive accuracy. On the other hand, mean squared error (MSE) measures the mean squared difference between predicted and actual values in a regression analysis, providing a measure of the predictive quality of the model. It is calculated by dividing the sum of squared differences between predicted and actual values by the number of observations. A lower MSE indicates a better fit between predicted and actual values and therefore higher forecast accuracy. Ultimately, an optimal prediction result would have a value of R^2 as close to 1 as possible and a value of MSE as small as possible.

It can be seen from Fig. 2 that for the three prediction methods of Transformer, RNN and LSTM, the prediction based on the reduced-order model has good performance, that is, it generally coincides with the real value, which shows that these three prediction methods are suitable, reasonable and valid for this case.

Further inspection of the graph reveals that it can be found that the predicted value and the actual value can basically fit and have the same trend in the stage of smoothing the curve, except for the larger gap in the RNN method, there is only a small gap between the remaining two algorithms. However, when the data fluctuates, the three methods more or less have some difference in trend between the predicted value and the actual value.

To further validate these observations, values of R^2 and MSE were computed for the three forecasting methods and are presented in tabular form in Fig. 2. The values of R^2 for Transformer, RNN, and LSTM are 0.7989, 0.4108, and 0.8268, respectively, while their corresponding values of MSE are 6.750e-5, 0.63143, and 4.906e-5, respectively.

The comparison of the values of MSE indicates that the LSTM and Transformer prediction methods have values on the same order of magnitude and are in proximity to zero. Thus, the predicted values for these two methods are relatively close to the actual value, with the differences being small. In other words, it means the squared differences between predicted and actual values distributed to each point are small. Moreover, the value of MSE of RNN is much higher than the other two, it can be seen that RNN is not suitable for this case.

Notably, the value of R^2 of the RNN algorithm is 0.4108, indicating that only 41.08 % of the variance in the dependent variable can be explained by the independent variables in the model. This finding is consistent with previous conclusions derived from the curves. For the value of R^2 of the Transformer and LSTM, the value of R^2 of LSTM is marginally better than that of the Transformer, and the prediction result of LSTM has the smallest value of MSE among the three. Therefore, for this data set and order reduction method, the LSTM algorithm seems to emerge as the optimal solution among the three.

In order to double verify the superior predictive performance of LSTM, it is necessary to replace the actual values used in the previous comparison with data that has undergone order reduction through PCA. This eliminates the possibility of any impact on predictive accuracy arising from alterations to the data integrity resulting from the use of PCA for order reduction.

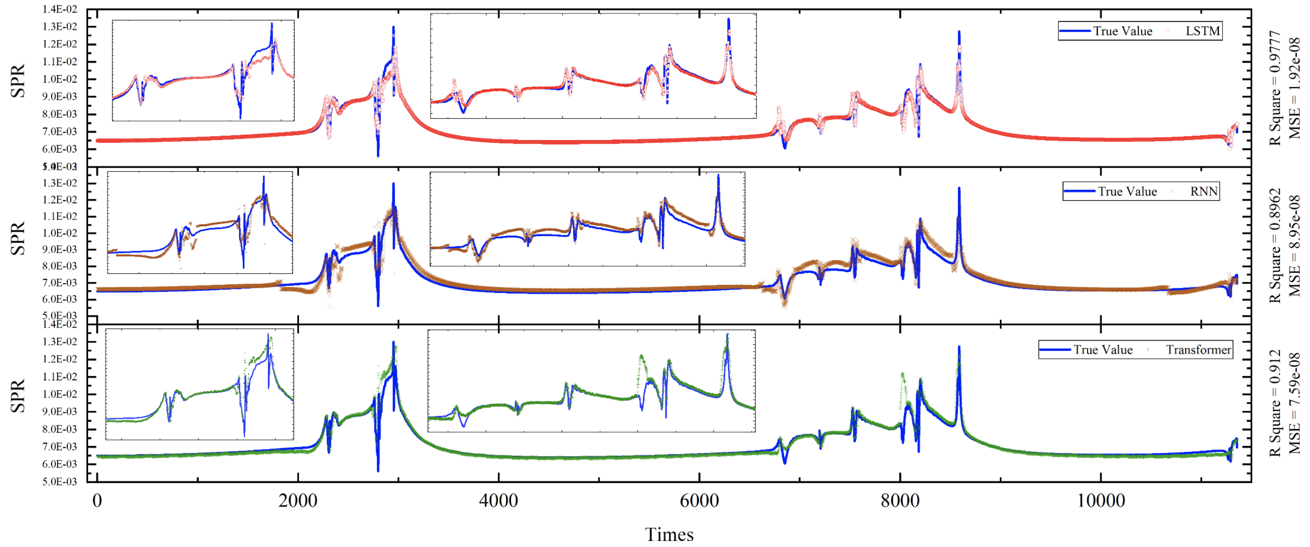


Fig. 3. Comparison of Transformer, RNN and LSTM in Predicted Value and Value of After Model Order Reduction

Based on the results presented in Fig. 3, it is evident that even when compared with the reduced data set, Long Short-Term Memory (LSTM) demonstrates the most superior performance in predicting values. Notably, the predictive accuracy compared with the value after model order reduction surpasses that when compared with the actual value. Thus, it is apparent that some of the deficiencies in the predictive ability of LSTM are attributable to the loss of data induced by PCA. Overall, it can be concluded that LSTM remains an excellent algorithm for prediction.

Overall, these findings suggest that while all three prediction methods demonstrate reasonable accuracy in forecasting, the LSTM algorithm offers the best prediction performance among the three, making it the most suitable choice for predicting continuous outcomes in similar scenarios.

Upon determining the optimal forecasting method for a given data set, the necessity of prior operations was validated utilizing the control variable approach. The methodology involved two distinct scenarios to ascertain the efficacy of the employed data preprocessing techniques, specifically, model order reduction by PCA and outlier removal, for enhancing the predictive accuracy of an LSTM model.

In the first test, the data set is trained with the LSTM model and compared the performance with and without outliers. Also, it should be noted that this set of tests did not use PCA for model order reduction. The results of the comparison revealed that the exclusion of outliers significantly enhanced the image-fitting degree of predicted values with actual values. The value of R^2 improved significantly from 0.0183 to 0.9819, and the mean squared error (MSE) decreased substantially from 0.0646 to 1.104e-5. These findings provide strong evidence supporting the correctness and necessity of the prior preprocessing operations.

In the second test, keep the condition of implementing outlier processing on the data set unchanged, then use LSTM to predict the model using PCA for order reduction and the model without order reduction. The comparison of the image fitting degree and associated performance metrics for the two models indicated that although there was not a too significant difference in the evaluation indicators, which are R^2 and MSE (With PCA: $R^2 = 0.8268$, MSE = 4.906e-5. Without PCA: $R^2 = 0.9819$, MSE = 1.104e-5), the training time varied substantially. Specifically, reducing the order of the data set significantly decreased the training time, with the PCA-reduced model requiring only 199.27s compared to 11873.31s for the non-reduced model. This finding underscores the importance of reducing the order of high-dimensional data sets to improve training efficiency and reduce computational complexity.

Therefore, the control variable approach provided valuable insights into the effectiveness of the preprocessing techniques applied to the data set. The findings demonstrate the importance of outlier removal and order reduction through PCA and their impact on improving the predictive accuracy and computational efficiency of the LSTM model.

Discussion of Performance of PCA. Although PCA is a widely used technique for reducing the order of high-dimensional data sets and satisfactory prediction results have been obtained in general when it works with LSTM, PCA downscaling to reduce training time comes at the expense of predictive accuracy. PCA is a linear technique, not perfect, and has several disadvantages that must be taken into account when using it for model order reduction.

Firstly, PCA does not preserve all the information in the original data set. In fact, it only preserves the information that is most relevant to the variance in the data. As a result, some of the less significant information in the original data set may be lost in the PCA-reduced dataset, which can potentially affect the accuracy of the model. This can be evidenced by some discrepancies between the predicted and actual values at the fluctuating curves. Also, by using Inverse PCA, also known as PCA reconstruction to recover the original data set from its model with dimensionality reduction obtained through PCA. Inverse PCA multiplies the reduced set of principal components by the transpose of the eigenvectors used to generate the principal components, the resulting matrix is then added to the mean of the original data set to obtain an approximation of the original data set. It can be found that the approximation is not exact since some information may have been lost during the PCA reduction process.

Moreover, PCA requires the computation of the eigenvectors and eigenvalues of the covariance matrix of the data set, which can be computationally expensive, particularly for large data sets. Moreover, the computation of eigenvectors and eigenvalues may also introduce numerical instability, leading to inaccuracies in the PCA-reduced data set.

In summary, while PCA is a useful technique for reducing the order of high-dimensional data sets, and provides good performance when it works with LSTM for prediction, it is not without its limitations. The potential loss of information and the computational expense are all factors that must be considered when using PCA for model order reduction. The presence of such risks poses a significant challenge to model reduction and prediction. It is important to note that while PCA may perform well in a specific case, caution must be exercised when attempting to generalize its performance to other scenarios. To optimize model reduction and make the reduced data have a better degree of restoration, another powerful technique called Dynamic Mode Decomposition (DMD) is worth considering. DMD is a nonlinear technique and can capture the underlying dynamics of the system more accurately, thus it may have a better performance than PCA. Details about the DMD technique will be discussed in the next chapter.

Dynamic Mode Decomposition. Dynamic Mode Decomposition (DMD) is an advanced mathematical algorithm that is frequently utilized to analyze and extract the underlying dynamics of a system from high-dimensional, time-varying data. DMD has been shown to be particularly effective in situations where large-scale datasets are available and the system of interest exhibits complex, nonlinear behavior. The numerous strengths of DMD, which coincide with the specific dataset used in this project, make it an ideal choice for the analysis of the system dynamics.[21]

Notably, DMD has a unique advantage over other commonly utilized techniques, such as PCA, as it does not result in significant loss of data during the working process. This means that the reduced order model generated by DMD can be effectively restored to the original dataset, which is not always possible with other techniques, such as PCA. As a result, DMD is a valuable tool for those seeking to obtain comprehensive and accurate insights into complex systems with high-dimensional, time-varying data.

Related Technique and Algorithm of DMD. DMD technique is rooted in the Singular Value Decomposition (SVD) of a data matrix, which enables the extraction of the dominant modes of behavior in a time-varying system. Specifically, DMD involves constructing a data matrix from a sequence of high-dimensional data snapshots, each representing the system at a distinct point in time. The data matrix is then structured such that each column represents a snapshot, and each row corresponds to a specific component of the system. Following this, the SVD of the data matrix is computed, resulting in the decomposition of the matrix into its constituent components. To proceed, DMD involves constructing two matrices, one of which is a low-dimensional approximation of the linear operator governing the system's temporal evolution, while the other represents the system's initial conditions. Subsequently, the eigendecomposition of the former matrix provides the DMD modes, which serve as dominant modes of behavior in the system. These modes can be utilized to predict future behavior, examine the system's stability, and facilitate an understanding of the underlying dynamics of the system.[22]

Moreover, DMD is closely related to the Koopman operator theory, which is a mathematical framework used to study the dynamics of nonlinear systems. Specifically, DMD can be viewed as a numerical approximation to the Koopman operator. The Koopman operator is an infinite-dimensional linear operator that describes the temporal evolution of observables in a nonlinear system. It provides a way to analyze nonlinear systems using linear tools, by describing the system's behavior in terms of the evolution of observable rather than the system's state itself. Specifically, DMD uses an approximation of the Koopman operator to analyze time-series data and extracts the dominant modes of behavior in the system. These modes can be used to predict future behavior, analyze the stability of the system, and gain insights into the underlying dynamics of the system.[22]

The following steps provide a more detailed illustration of the algorithm of DMD.

Step 1. Acquire pairs of system state snapshots as it progresses through time:

$$\left\{x\left(t_k\right), x\left(t'_k\right)\right\}_{k=1}^{m} \quad [8]$$

$$t'_k = t_k + \Delta t \quad [9]$$

Step 2. Organize the collected data into two matrices:

$$X = \left[x\left(t_1\right) x\left(t_2\right) ... x\left(t_m\right)\right] \quad [10]$$

$$X' = \left[x\left(t'_1\right) x\left(t'_2\right) ... x\left(t'_m\right)\right] \quad [11]$$

Step 3. Determine the optimal linear operator A, which establishes the relationship between the two matrices of system state snapshots, X and X':

$$X' = AX \quad [12]$$

where A is the arguments of the minimum of the Frobenius norm of X'-AX, which equals to the product of X' and the pseudo-inverse of X.

Step 4. Compute the SVD of X:

$$X \approx \widetilde{U}\widetilde{\Sigma}\widetilde{V}^* \quad [13]$$

The choice of the rank r for SVD is one of the most important steps of the DMD.

Step 5. Compute the matrix A using the SVD computed at the previous step:

$$A = X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^* \quad [14]$$

$$\tilde{A} = \tilde{U}^* A \tilde{U} = \tilde{U}^* X' \tilde{V} \tilde{\Sigma}^{-1} \quad [15]$$

Step 6. The spectral decomposition of the reduced matrix of A can be computed as:

$$\tilde{A}W = W\Lambda \quad [16]$$

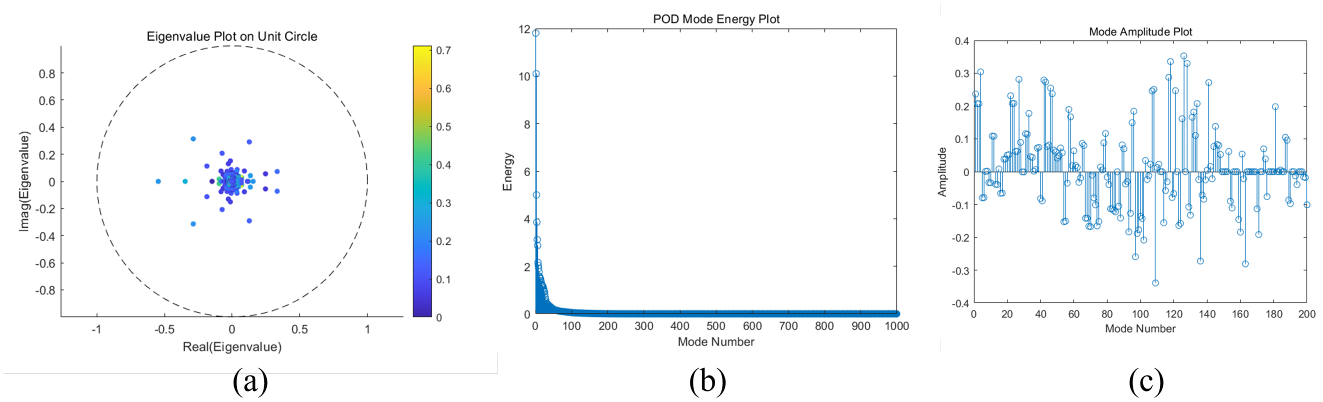
Step 7. Compute the high-dimensional DMD modes are reconstructed using the eigenvectors W of the reduced system and the time-shifted snapshot matrix X':

$$\Phi = X'\tilde{V}\tilde{\Sigma}^{-1}W \quad [17]$$

Where the DMD modes Φ are eigenvectors of the high-dimensional matrix corresponding to the eigenvalues in A

Based on the above steps, DMD technology can be used for model reduction and prediction. The third step can resemble the Koopman operator, and the rank of the data matrix is 200 which is used for SVD in step 4.

Dynamic System Analysis. The goal of dynamic system analysis is to identify patterns in the behavior of a system over time and to understand the factors that influence these patterns. This can involve examining the system's response to external stimuli, the stability of its dynamics, and the presence of any underlying trends or cycles.



(a) (b) (c)

Fig. 4. Dynamic System Analysis

In the context of DMD, the eigenvalues lying on the unit circle represent the stability and periodicity of the underlying dynamical system. By decomposing the system matrix, a set of eigenvectors and eigenvalues can be obtained. Typically, these eigenvalues are depicted on the complex plane and may reside either inside or outside the unit circle.

The (a) in Fig. 4 shows that all eigenvalues are confined within the unit circle, indicating that the system is stable and attenuated. This implies that any disturbances introduced into the system will eventually decay, thereby validating its predictability. Furthermore, the eigenvalues concentrate near the center of the unit circle, indicating that the system's dynamic behavior is predominantly governed by low-frequency modes, corresponding to the steady-state or slowly changing the behavior of the system. These low-frequency modes, which capture the long-term dynamic behavior of the system, can be extracted by DMD and utilized to establish a simplified model for predicting and controlling the system's long-term behavior. It is worth noting that the absence of eigenvalues outside the unit circle suggests that the system is unlikely to exhibit complex chaotic behavior.

Furthermore, DMD can effectively decompose the dynamic behavior of a system into a set of dynamic modes, each corresponding to a specific vibration frequency and amplitude. The energy associated with these modes can be quantified by the magnitudes of the corresponding eigenvalues. Hence, the mode and energy diagram of DMD, depicted in (b) in Fig. 4 as the energy-mode diagram, carries great significance in comprehending the vibration characteristics of the system, the coupling relationships between modes, and the energy distribution of modes. The energy-mode diagram displays the energy values of all modes in the system and serves as a crucial tool for extracting the dimensions and characteristics of the reduced-order system. Based on the energy analysis of each mode and their respective order of magnitude, it is observed that the first 200 modes account for most of the system's energy. As such, the dimension of singular value truncation can be set to 200, reducing the system's dimensionality from 2560 to 200. This approach not only simplifies the system but also ensures that the first 200 modes accurately capture the essential features of the original system. Consequently, the reduced-order system can be efficiently restored to the original system with minimal information loss.

Moreover, the (c) in Fig. 4 exhibits the amplitude associated with each mode and highlights their respective contributions to the system. By analyzing the mode diagram, the primary dynamic modes can be identified, leading to a deeper comprehension of the dynamic behavior and characteristics of the system.

Comparison the Performance of DMD and LSTM with PCA. For model order reduction, while PCA and DMD are both techniques for dimensionality reduction, they differ in their focus and applications. PCA aims to transform high-dimensional data into a new set of orthogonal components, known as principal components, which capture the most significant variations in the data. In contrast, DMD is a data-driven method that decomposes dynamic systems into a set of spatiotemporal modes based on their temporal evolution. Therefore, PCA is primarily suited for static data analysis, while DMD is more applicable in dynamic systems analysis.

For analyzing time-series data, DMD and LSTM can both analyze and predict time-series data. DMD requires a finite amount of data and can provide a simplified model that captures the essential features of the system. LSTM is a non-linear model and can learn complex patterns in data, making it suitable for data that contains non-linear dynamics.

As previously noted in the preceding section, PCA is associated with a substantial loss of data, which may impede the ability to recover the original data and adversely impact the performance of LSTM models. In the other words, there is a conspicuous disparity between the predicted values obtained from LSTM training and the actual values acquired through PCA for model order reduction. This is the reason why this project has implemented a hybrid approach that combines PCA and LSTM for data analysis and prediction, while still utilizing DMD for model order reduction and prediction to ameliorate these shortcomings. By adopting this method, the chapter aims to achieve a more comprehensive understanding of the underlying dynamics of the data, facilitate accurate predictions, and minimize information loss. In order to better compare the performance of DMD and PCA+LSTM, Fig. 5 is drawn.

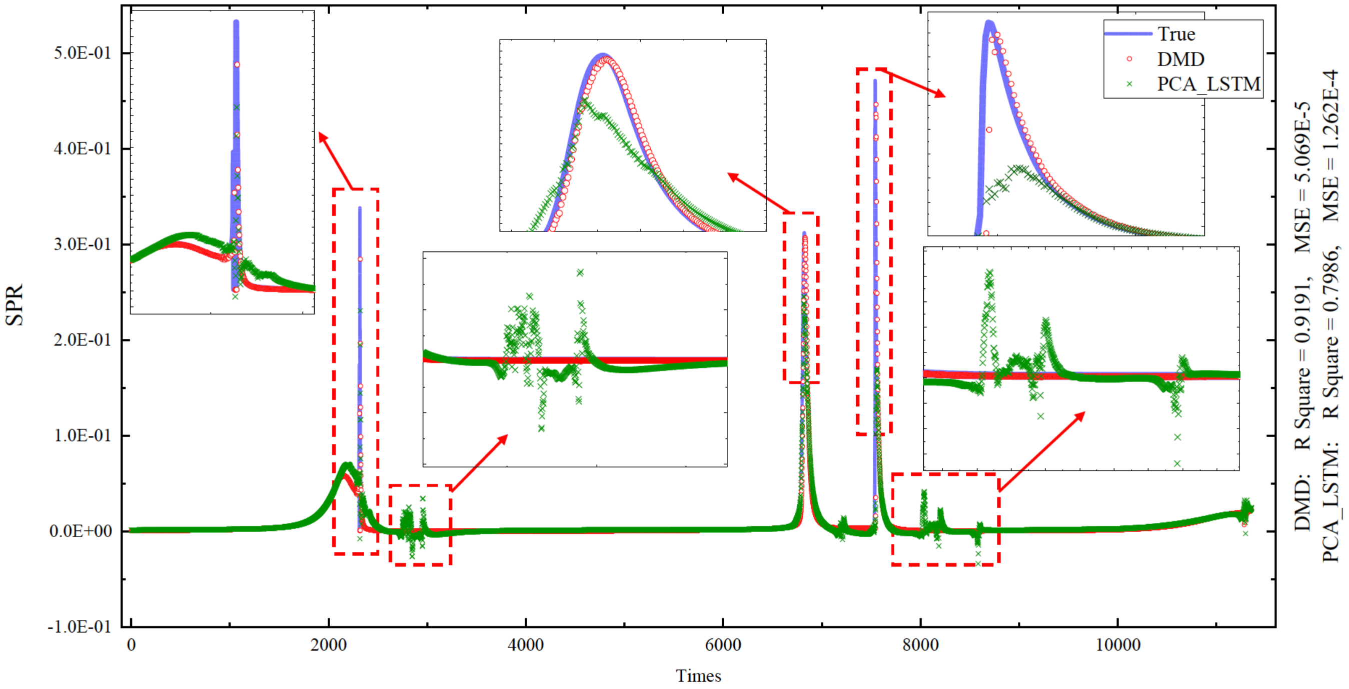


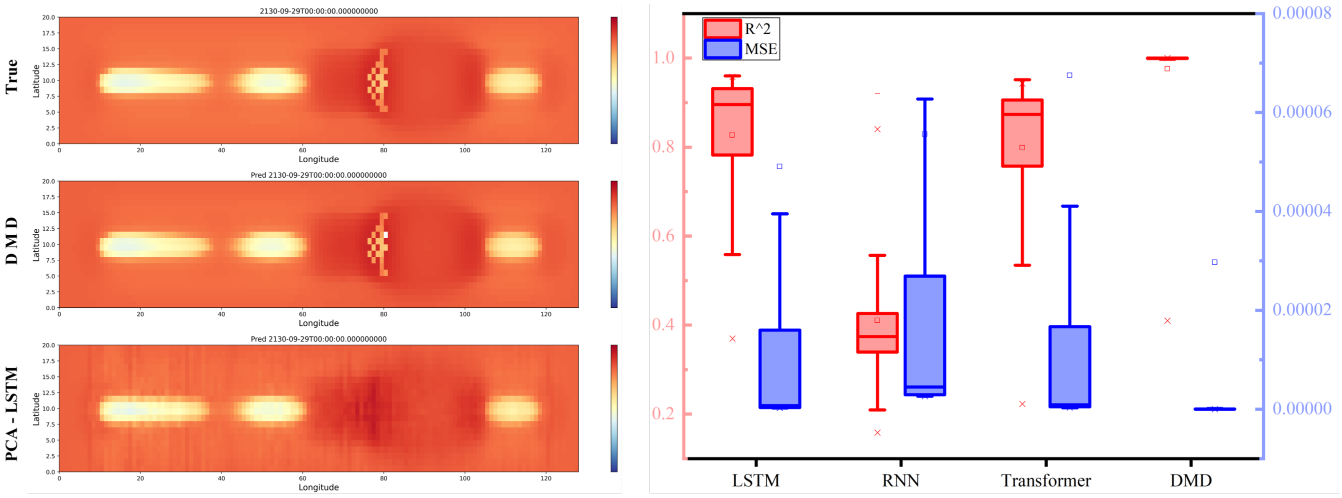
Fig. 5. Comparison the Performance of DMD and LSTM with PCA

Based on the results shown in Fig. 5, both DMD and the combination of PCA and LSTM demonstrate satisfactory performance in the smooth part of the curve. However, noticeable discrepancies arise between the two methods in the vicinity of the peak and surrounding data, with DMD exhibiting superior performance. Specifically, the predicted values obtained from the PCA and LSTM combination tend to deviate significantly from the actual values at the peak due to the substantial data loss incurred during PCA-based model order reduction. In contrast, the predicted values obtained from DMD training effectively capture the peak value and align with the actual value. Additionally, the prediction results derived from the PCA and LSTM combination exhibit fluctuation before and after the peak, indicating a slight reduction in stability compared to DMD. Overall, these observations highlight the robustness of DMD in accurately predicting system behavior and underscore the limitations of PCA-based approaches in preserving critical information for accurate predictions.

Furthermore, in order to derive more precise and rigorous conclusions regarding the comparative performance of the prediction methods, two key parameters, R and MSE, were used to evaluate their predictive capabilities. For the DMD method, the value of R is 0.9191, and the value of MSE is 5.069e-5. As for the PCA+LSTM method, the value of R is 0.7986, and the value of MSE is 1.262e-4. As explained in the previous section, the effectiveness of a prediction model is determined by its ability to achieve a value of R close to 1 and an MSE value as small as possible. Therefore, based on these parameters, it can be concluded that the prediction performance of DMD surpasses that of PCA+LSTM in this dataset. Additionally, the training time of both methods is not significantly different, and it is worth noting that the good training time achieved by PCA+LSTM comes at the expense of reduced prediction accuracy.

Based on the above analysis, it can be obtained that the DMD algorithm exhibits superior prediction performance compared to the PCA+LSTM method. This highlights the fact that PCA, as discussed earlier, only retains the most significant information related to the variance of the original dataset, leading to the loss of certain data during the model order reduction process. This loss of information ultimately affects the accuracy of the model, as demonstrated by the differences in prediction accuracy between the two methods. Consequently, previously ignored data, thought to be insignificant, are shown to be crucial and have a significant relationship with the model's characteristics. DMD is capable of more accurately capturing the underlying dynamics of the system and achieving a higher degree of reduction in the reduced data. The predicted value obtained by DMD not only can restore the original data but also have better predictive performance. Therefore, adopting the DMD method to optimize the prediction performance after using PCA+LSTM to achieve good prediction results is a sound approach.

In order to assess the stability of the DMD and PCA+LSTM methods, a commonly used graphical tool known as the box plot is employed. The box plot is a statistical representation of a dataset that provides a summary of its distribution. In the context of stability analysis, a box plot can be used to visualize the distribution of the eigenvalues of a system on the complex plane. The interquartile range, which contains the middle 50% of the eigenvalues, is represented by the box portion of the plot, with the median value of the eigenvalues depicted as a line within the box. A box plot is a useful tool for quickly evaluating the stability and frequency behavior of a dynamical system, as the height of the box indicates the concentration of the data and the overall stability of the system. Figure 6 displays the box plots of DMD and PCA, which provide insight into the stability performance of each method.



(a) (b)

Fig. 6. (a)Comparison of Inversed PCA and DMD. (b)The Box Plots of DMD and PCA

Fig. 6 (a) visulize the spatial status of the system at specific time after inversing the reduced order system. It is obvious that DMD has a better capability to restore the system, which means that DMD can retain the majority information of system.The box plot depicted in Fig. 6 (b) provides an evident comparison between the stability of the DMD method and the three prediction algorithms based on the PCA method. Notably, the height of the box plot for the DMD method is considerably lower than that of the PCA-based algorithms, and the box is almost a line, indicating a more stable dynamic system. This suggests that the DMD algorithm is more efficient in reducing the model order while retaining the essential characteristics of the system, thereby improving its stability. Therefore, the DMD method is an effective tool for system identification and model order reduction, which can ultimately enhance the prediction accuracy of the system.

Conclusion. In conclusion, the comparison between the Dynamic Mode Decomposition (DMD) and PCA+LSTM methods reveals that DMD outperforms PCA+LSTM in terms of prediction accuracy and stability. The results indicate that PCA-based model order reduction incurs significant data loss, particularly in the vicinity of peaks and critical data points, leading to reduced prediction accuracy. In contrast, the DMD method more accurately captures the system dynamics and achieves a higher degree of reduction in the reduced data while preserving critical information. Furthermore, the box plot analysis confirms that the DMD method exhibits superior stability compared to the PCA-based algorithms. Thus, the DMD algorithm is an effective tool for system identification and model order reduction, which can ultimately enhance the prediction accuracy of the system. The findings of these tests show that the DMD algorithm is efficient and suitable for various fields, including engineering, physics, and finance, where accurate predictions and stable system behavior are crucial.

ACKNOWLEDGMENTS. Sincerely appreciate Prof. Gianmarco Mengaldo and Dr.Adriano Gualandi for the wonderful and meaningful lectures. We do learn a lot from the lectures. Furthermore, we can make what have learned into practice in this project, which makes us impressive. Hoping this teaching method and style can insist, and instruct more and more NUS students to learn further in data-driven engineering and machine learning.

References

[1] Lin Li et al. “Soil seismic response modeling of KiK-net downhole array sites with CNN and LSTM networks”. In: *Engineering Applications of Artificial Intelligence* 121 (May 2023), p. 105990. DOI: 10.1016/j.engappai.2023.105990. (Visited on 02/28/2023).

[2] Laura Laurenti et al. “Deep learning for laboratory earthquake prediction and autoregressive forecasting of fault zone stress”. In: *Earth and Planetary Science Letters* 598 (Nov. 2022), p. 117825. DOI: 10.1016/j.epsl.2022.117825. (Visited on 04/08/2023).

[3] Ewnetu Abebe et al. “Earthquakes magnitude prediction using deep learning for the Horn of Africa”. In: *Soil Dynamics and Earthquake Engineering* 170 (July 2023), p. 107913. DOI: 10.1016/j.soildyn.2023.107913. (Visited on 04/08/2023).

[4] Luca Franco et al. “Under the hood of transformer networks for trajectory forecasting”. In: *Pattern Recognition* 138 (June 2023), p. 109372. DOI: 10.1016/j.patcog.2023.109372. (Visited on 04/08/2023).

[5] Bo Zhang et al. “EPT: A data-driven transformer model for earthquake prediction”. In: *Engineering Applications of Artificial Intelligence* 123 (Aug. 2023), p. 106176. DOI: 10.1016/j.engappai.2023.106176. (Visited on 04/06/2023).

[6] Qilin Li et al. “A comparative study on the most effective machine learning model for blast loading prediction: From GBDT to Transformer”. In: *Engineering Structures* 276 (Feb. 2023), p. 115310. DOI: 10.1016/j.engstruct.2022.115310. (Visited on 04/08/2023).

[7] Ce JIANG et al. “Comparison of the Earthquake Detection Effects of PhaseNet and EQTransformer considering the Yangbi and Maduo Earthquakes”. In: *Earthquake Science* 34 (2021), pp. 1–11. DOI: 10.29382/eqs-2021-0038. (Visited on 04/11/2022).

[8] Xiaotao Chang et al. “One sliding PCA method to detect ionospheric anomalies before strong Earthquakes: Cases study of Qinghai, Honshu, Hotan and Nepal earthquakes”. In: *Advances in Space Research* 59 (Apr. 2017), pp. 2058–2070. DOI: 10.1016/j.asr.2017.02.007. (Visited on 04/08/2023).

[9] Jyh-Woei Lin. “Potential reasons for ionospheric anomalies immediately prior to China’s Wenchuan earthquake on 12 May 2008 detected by nonlinear principal component analysis”. In: *International Journal of Applied Earth Observation and Geoinformation* 14 (Feb. 2012), pp. 178–191. DOI: 10.1016/j.jag.2011.09.011. (Visited on 01/29/2022).

[10] Qiugang Lu, Sungho Shin, and Victor M. Zavala. “Characterizing the Predictive Accuracy of Dynamic Mode Decomposition for Data-Driven Control”. In: *IFAC-PapersOnLine* 53 (2020), pp. 11289–11294. DOI: 10.1016/j.ifacol.2020.12.373. (Visited on 01/05/2023).

[11] Matthew O. Williams et al. “Extending Data-Driven Koopman Analysis to Actuated Systems”. In: *IFAC-PapersOnLine* 49 (2016), pp. 704–709. DOI: 10.1016/j.ifacol.2016.10.248. (Visited on 11/10/2020).

[12] Seth M. Hirsh, Bingni W. Brunton, and J. Nathan Kutz. “Data-driven spatiotemporal modal decomposition for time frequency analysis”. In: *Applied and Computational Harmonic Analysis* 49 (Nov. 2020), pp. 771–790. DOI: 10.1016/j.acha.2020.06.005. (Visited on 12/03/2021).

[13] Aleksandr Katrutsa, Sergey Utyuzhnikov, and Ivan Oseledets. “Extension of Dynamic Mode Decomposition for dynamic systems with incomplete information based on t-model of optimal prediction”. In: *Journal of Computational Physics* 476 (Mar. 2023), p. 111913. DOI: 10.1016/j.jcp.2023.111913. (Visited on 04/08/2023).

[14] Bingni W. Brunton et al. “Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition”. In: *Journal of Neuroscience Methods* 258 (Jan. 2016), pp. 1–15. DOI: 10.1016/j.jneumeth.2015.10.010. URL: https://arxiv.org/pdf/1409.5496.pdf (visited on 01/04/2022).

[15] Mohamed A. Abu-Seif et al. “Data-Driven modeling for Li-ion battery using dynamic mode decomposition”. In: *Alexandria Engineering Journal* 61 (Dec. 2022), pp. 11277–11290. DOI: 10.1016/j.aej.2022.04.037. (Visited on 01/05/2023).

[16] Ian T. Jolliffe and Jorge Cadima. “Principal component analysis: a review and recent developments”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374 (Apr. 2016), p. 20150202. DOI: 10.1098/rsta.2015.0202.

[17] Wilhelmus H Schilders et al. *Model Order Reduction: Theory, Research Aspects and Applications.* Springer Berlin Heidelberg, 2008, pp. 3–32.

[18] Zu-Qing Qu. *Model Order Reduction Techniques with Applications in Finite Element Analysis.* Springer Science Business Media, Mar. 2013, pp. 1–11.

[19] Tommaso Boccato, Alberto Testolin, and Marco Zorzi. “Learning Numerosity Representations with Transformers: Number Generation Tasks and Out-of-Distribution Generalization”. In: *Entropy* 23 (July 2021), p. 857. DOI: 10.3390/e23070857. (Visited on 06/15/2022).

[20] Alex Sherstinsky. “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network”. In: *Physica D: Nonlinear Phenomena* 404 (Mar. 2020), p. 132306. DOI: 10.1016/j.physd.2019.132306.

[21] Jose Manuel Vega and Soledad Le Clainche. *Higher Order Dynamic Mode Decomposition and Its Applications.* Academic Press, Sept. 2020, pp. 29–83.

[22] PETER J. SCHMID. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (July 2010), pp. 5–28. DOI: 10.1017/s0022112010001217.