IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 November 2023, accepted 14 December 2023, date of publication 21 December 2023,
date of current version 3 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3345456

APPLIED RESEARCH

Open Set Domain Adaptation for Classification of Dynamical States in Nonlinear Fluid Dynamical Systems

S. AKSHAY[1], E. A. GOPALAKRISHNAN[2,3], V. SOWMYA[1], J. VENKATRAMANI[4], DHEERAJ TRIPATHI[4], JAY SHANKAR PRASAD[5], AND SIRSHENDU MONDAL[5]

[1]Center for Computational Engineering and Networking (CEN), Amrita Vishwa Vidyapeetham, Coimbatore 641112, India
[2]Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru 560035, India
[3]Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Bengaluru 560035, India
[4]Department of Mechanical Engineering, Shiv Nadar IoE, Gautam Buddha Nagar, Uttar Pradesh 201314, India
[5]Department of Mechanical Engineering, National Institute of Technology Durgapur, Durgapur 713209, India

Corresponding author: E. A. Gopalakrishnan (ea_gopalakrishnan@blr.amrita.edu)

The work of E. A. Gopalakrishnan and J. Venkatramani was supported by the Science and Engineering Research Board (SERB), India, under Grant CRG/2022/001609.

ABSTRACT Nonlinear fluid dynamical systems, such as thermoacoustic systems, aeroelastic systems are archetypical complex systems involving state transitions upon a change in bifurcation parameter. These state transitions in any certain direction are always undesirable and can radically alter the operational paradigms associated with these systems. Hence, predicting the impending dynamical state is paramount for avoiding such undesirable transitions. The hitherto research so far focused largely on metric-based and model-based indicators to foretell an impending transition and is often fraught with difficulties when deployed in practicable scenarios. In this study, we assuage this end of concern by proposing a model-agnostic data-driven method for automated classification of the dynamical states of nonlinear fluid dynamical systems. By using recurrence plots we transform the time series pertaining to the dynamical states into images and subsequently employ a convolution neural network (CNN) to classify the generated images. This study also proceeds to present cross-domain classifications via a trained deep learning (DL) model and successfully classify the dynamical states of one fluid dynamical system (say, thermoacoustic) with the dynamical states of another fluid dynamical system (say, aeroelastic). The underlying methodology for the above is based on open set (OS) domain adaptation - inherent to transfer learning schemes. Towards enhancing the confidence levels of our proposed methodology, we carry out four cross-domain numerical experiments, wherein we consistently get about 94 - 98% accuracy.

INDEX TERMS Dynamical systems, deep learning, open set domain adaptation, pattern recognition, recurrence plots.

I. INTRODUCTION

The study of nonlinear dynamical systems is crucial for deepening our understanding of the physical world. Different dynamical systems display varying complexity and diverse behavior far from equilibrium states. One crucial aspect of dynamical systems is state transitions. State transitions in dynamical systems can occur in one of two ways: through a gradual change in the system parameters [1] or due to external fluctuations [2]. The former is typically termed bifurcations, and the latter is called noise-induced transitions. Bifurcation theory is used to study these transitions. There are several types of bifurcations, such as fold bifurcation in a pitch-plunge aeroelastic system exhibiting stall flutter [3], [4], Hopf bifurcation in aeroelastic systems in the presence of cubic hardening nonlinearity [5], and transcritical bifurcation

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar.

in an asymmetric 2D pitching wing section with cubic stiffness [6]. Each bifurcation type corresponds to distinct dynamical behavior or state.

Identifying different dynamical states is paramount in predicting transitions in dynamical system analysis. To date, empirical indicators based on the system's time series have been used to predict impending transitions. Metric-based and model-based indicators are the two broad categories to which these indicators belong. Model-based indicators involve fitting a specific model to the data, while metric-based indicators assess changes in the system's behavior without linking the data to a specific model. Prominent metric-based indicators include the Largest Lyapunov exponent (LLE) [7] and Shannon entropy [5], [8], [9], while examples of model-based indicators are time-varying AR(p) models [2] and non-parametric drift-diffusion-jump models [1]. However, these methods have limitations in practical applications, such as the requirement for noise-free and lengthy time series.

The disproportionate emphasis placed on metric-based or model-based quantification in fluid dynamical systems [5], [10], [11], [12] is a concern with the available methodologies for predicting state transitions. The interaction of fluid flow with flexible structures or acoustic-heat sources can be best described by the paradigms of nonlinear dynamical systems. Aeroelasticity, which refers to the interaction of fluid flow with structures, and thermoacoustic behavior, which involves the interaction of flows with heat-acoustic sources, are prone to unpredictable and often catastrophic bifurcations [3], [4], [10], [11], [13]. Interestingly, these two fluid dynamical systems can exhibit similar bifurcation scenarios under specific conditions. A key concern with these catastrophic transitions in response dynamics is the sudden emergence or disappearance of oscillatory instabilities. For example, the aeroelastic system can give rise to self-sustained limit cycle oscillations (LCOs) in the structural response, known as flutter, at a critical bifurcation parameter. Similarly, a thermoacoustic system can exhibit sustained pressure oscillations called thermoacoustic instability, which also manifests as LCOs in the pressure responses [13], [14], [15].

The two fluid dynamical systems considered in this study are ubiquitous in various applications, ranging from power and propulsion devices [16] to lifting surfaces such as aircraft wings, micro aerial vehicles (MAVs), and wind turbine blades [17]. Indeed, both systems exhibit their primary instability through a Hopf bifurcation under deterministic conditions and demonstrate an intermittency route to Limit Cycle Oscillations (LCOs) under turbulent conditions [13], [14].

Pulse combustors belong to a category of air-breathing engines that utilize combustion pulsations to improve their performance. Traditional pulse combustors use either mechanical or aerodynamic valves to regulate the intake of air and fuel, which is determined by the pressure within the combustion chamber. However, thermal pulse combustors exhibit pulsating combustion even when there is a constant inflow of air and fuel across various operating conditions. This pulsation arises due to complex interactions between the system's acoustic field and the irregular heat release from combustion, leading to thermoacoustic instabilities. These interactions result in intricate system dynamics, including periodic, quasiperiodic, period doubling, and chaotic oscillations. Factors like wall temperature and the air-fuel ratio influence the system's behaviour [10], [11], [18], [19], [20], [21], [22].

On the other hand, nonlinear aeroelastic systems can generate various bifurcations and unpredictable instabilities. The type and source of nonlinearity in an aeroelastic system significantly influence the aeroelastic response dynamics and give rise to different bifurcation routes. For example, a cubic hardening nonlinearity in structural stiffness, caused by large deformations, results in a supercritical Hopf bifurcation [14]. Structural freeplay nonlinearity, which arises from loose hinges or worn parts, leads to a period-doubling route to chaos and subsequent divergent oscillations [23]. Aerodynamic nonlinearity induced by dynamic stall can lead to both subcritical [3], [4], [24] and supercritical Hopf bifurcation routes [25], as well as a period-doubling route to chaos [23], [26].

There has been a growing concern in industries that deploy fluid dynamical systems such as gas turbine combustors, wind turbine blades, aircraft wings, micro aerial vehicles (MAVs), etc. The sudden onset of large amplitude Limit Cycle Oscillations (LCOs) and subsequent secondary bifurcations is considered detrimental to structural integrity [16], [17]. From a structural safety perspective, these LCOs and their counterparts, such as period-doubling cascades, can result in (i) abrupt structural failure upon the first occurrence [17] or (ii) significant fatigue damage accumulation due to stress cycle reversals [27], [28], [29]. Consequently, the structural safety of these fluid dynamical systems is compromised by the sudden onset of these dynamical signatures. In response, both the aeroelastic and thermoacoustic communities have invested considerable efforts in predicting, preventing, and suppressing instabilities in their respective fluid dynamical systems [5], [12], [13], [14], [18], [19], [20], [26], [30]. However, in the case of pulse combustors, a type of thermoacoustic system, the intention is to operate with oscillatory combustion [10], [11], [21], [22], [31]. Therefore, the primary focus is on controlling and maintaining oscillatory combustion rather than suppressing it in pulse combustors. Consequently, predicting the dynamical state transitions remains crucial, even with the intent to sustain and control the oscillatory combustion.

Additionally, both the aeroelastic and thermoacoustic problems are highly complex multiphysics problems, wherein the role of nonlinearity, fluid characteristics, and the sensitivity to parameter degradation is not fully understood [3], [5], [10] and, in turn, compelling the community to formulate alternate metrics to foretell oscillatory instabilities. In other words, the suite of measures, such as, say,

characterizing multifractality [12], [30] or using a complexity measure [5], is tailored for specific bifurcation scenarios. However, discontinuous nonlinearities (for example, freeplay in the structural stiffness) or stochasticity in the problem can jeopardize traditionally known bifurcation routes and can give rise to atypical bifurcations and, in turn, rendering the metric-based and model-based measures ineffective. This study focuses on addressing the need for developing a data-driven approach to classify dynamical signatures. The main objective is to create an AI-based model-agnostic tool capable of understanding and recognizing patterns in time series data, which is crucial for characterizing and preventing undesirable transitions. It's important to clarify that our intention is not to propose an alternative method to the existing traditional state transition prediction techniques in nonlinear dynamics.

In this paper, we present an automated deep learning approach for classifying the dynamical states of nonlinear fluid dynamical systems. We utilize a convolutional neural network (CNN) to process image representations of time series data. Specifically, we transform the time series related to a dynamical state of a nonlinear fluid dynamical system into a recurrence plot (RP). These generated recurrence plots serve as input to the CNN for classification purposes.

Given that many nonlinear fluid dynamical systems can exhibit similar bifurcation scenarios, this work also introduces the concept of cross-domain dynamical states classification using open-set domain adaptation (OS). Open-set domain adaptation is a technique used to handle situations where the source and target domains have inherent differences, making it relevant for our research in classifying dynamical states across different scenarios.

Through OS, a model trained to classify dynamical states for a specific nonlinear fluid dynamical system can be extended to classify dynamical states of other nonlinear fluid dynamical systems after making necessary adjustments to the deep learning model. This approach eliminates the need to train a new model for each nonlinear fluid dynamical system. In a practical case, the unknown class typically outnumbers the known classes, making it challenging to predict unknown classes with high precision. To overcome this issue, we train models with unknown class samples along with known classes of the source domain and increase the individual class accuracy. This also implies that each time-series image must have its class labelled, which is quite challenging in real-world situations. In this work, we conducted four cross-domain experiments. Our proposed work yields test accuracy of 98% in experiment 1, 97% in experiment 2, 96% in experiment 3, and 97% in experiment 4. To our understanding, this study represents the first attempt to utilize OS domain adaptation and 2D CNN for the classification of dynamical states in nonlinear fluid dynamical systems.

The following steps constitute this research paper's workflow.

- First, we obtain time series sequence data pertaining to each dynamical state of those aforementioned two nonlinear fluid dynamical systems.
- Second, a recurrence plot is generated for each time series in the dataset. These images are then given as input to the CNN-based DNN (Deep Neural Network) classifier.
- Third, the OS domain adaptation experiment is done for 4 source and target domain combinations. Table 2 represents the experiments done in this proposed work. It should be noted that the CNN classifier architecture is kept constant for all 4 experiments (Table 7). For the purpose of performance validation, we use three performance evaluation metrics/measures. Figure 2 shows the pipeline of this study.

The rest of the article is organised as follows. Section I-A presents related works discussion. In section II, we briefly explain the mathematical models of the fluid dynamical systems considered in this study. In section III, we detail the dataset used in this study. Second, we discuss the generation of recurrence plots from the time series data. Finally, we describe the CNN model as well as its parameter choices. Section IV presents the results and discussions. Finally, section V presents the major conclusions of this study.

A. RELATED WORKS

1) DEEP LEARNING LITERATURE REVIEW

Deep learning (DL) falls within the realm of machine learning as a specialized sub-field, which was first suggested by Hinton et al. [32] in 2006. The idea behind deep learning is to superimpose several hidden layers to mimic the human brain's neural network to learn new things. Deep learning is one such technique that has several benefits over other pattern recognition techniques. Deep learning models have achieved cutting-edge performance in computer vision [33], [34], [35], speech recognition [36], [37], and time series classification [38], [39], [40], [41], [42], [43]. In contrast to traditional machine learning algorithms, DL algorithims such as Convolutional Neural Networks (CNN) with their hierarchical representation, feature learning capabilities, and adaptability to unknown data, stand as a fitting choice for achieving our research objectives.

Convolutional Neural Networks (CNN) are among the most widely utilized Deep Learning (DL) techniques in the fields of computer vision and image analysis [44]. Unlike the traditional classification framework with hand crafted-features, they are highly noise-resistant models and can extract highly informative, deep features with little to no data feature engineering. Feature learning and classification are seamlessly integrated into a single model and are co-learned, resulting in mutual performance enhancement. This is achieved through the utilization of multiple hidden layers comprising various processing units, including convolution, pooling, ReLU (rectified linear unit), and normalization.

These layers collectively learn a hierarchy of features, progressing from lower-level to higher-level representations. Several variants of CNN have since been proposed and are employed in various application sectors. Boullé et al. proposed CNN to generalize and classify chaotic time series with high accuracy and observed that CNN without batch normalization outperforms state-of-the-art neural networks in time series classification [45]. Chen et al. proposed CNN to classify mechanical fault types in two datasets: gearbox and motor bearing [46]. Barros and Ebecken proposed CNN to classify ships based on acoustic signatures [47]. CNN was proposed by Ben-Cohen et al. as a technique for classifying CT images of the liver [48]. CNN have been utilized for other applications such as image spam classification [49], facial emotion classification [50], and fingerprint liveness detection system [51]. Therefore, in the present work, the ubiquitous influence of CNN in image classification applications serves as an impetus to utilize DL in classifying the dynamical states of the considered fluid dynamical systems.

2) CONVOLUTIONAL NEURAL NETWORK (CNN) AND RECURRENCE PLOT

Recurrence plots offer a means to visually depict the periodic characteristics of a trajectory through a phase space and enable us to investigate particular attributes of the m-dimensional phase space trajectory through a 2D representation [52]. Therefore, representing time series data as images introduces unique features that are not present in 1D signals. This transformation allows time series classification to be approached as an image recognition task. The generated recurrence plots are then inputted to CNN for classification in this work. In hitherto literature, CNN and RP have been successfully used to analyze time series data. Mukhopadhyay et al. transformed time series data from chaotic dynamical systems such as Mackey-Glass, Lorenz, Rössler, and Duffing oscillators into image representations such as un-thresholded recurrence plots, which are fed into a CNN to classify different system dynamics such as noisy chaos, clean deterministic chaotic and stochastic dynamics [53]. Lee et al. transformed capacity degradation data of Li-ion batteries into recurrence plot and Gramian angular fields, which are then fed into five types of CNN's models for state-of-health estimation [54]. Zhao et al. used CNN to classify foetal heart rate (FHR) signals from a recurrence plot to forecast foetal hypoxia [55]. Arratia and Sepúlveda predicted financial time series data for the Standard & Poor's 500 index and bank using CNN and recurrence plots [56]. Recurrence plots were utilized by Chen et al. to transform financial time series into images, which they then used to forecast trends in financial data [57]. Therefore, the efficacy of recurrence plots in capturing essential patterns and information from a time series data combined with CNNs effectiveness in extracting relevant features from images and learning hierarchical features provides an effective platform for classifying/detecting different dynamical states of nonlinear fluid dynamical systems.

3) OPEN SET DOMAIN ADAPTATION LITERATURE REVIEW

Deep learning (DL) algorithms require a large amount of data, making them data-intensive and necessitating thousands of time series for effective classification. However, in the literature, DL models proposed for canonical dynamical systems pose challenges when dealing with real-time systems or models. Furthermore, a DL model designed for one system's dynamical states may not be transferable to others. To address this, transfer learning techniques (TL), including open set domain adaptation (OS), have been proposed to improve model generalizability across diverse domains. Traditional machine learning assumes training and test sets from similar distributions. Yet, real-world situations often feature differing distributions due to various sources or data evolution. Blindly applying a trained model to a new dataset can lead to reduced performance on the test data.

Domain adaptation is a component of Transfer learning (TL) [58], [59], [60] that seeks to deal with these types of issues by mitigating the difference between domains so that the trained model can be generalised into the domain of interest. In this context, the training and test sets are designated as the source and target domains, respectively. Domain adaptation encompasses two distinct categories: closed-set (CS) and open-set (OS). When the source and target domains have an equal number of classes, CS domain adaptation is employed. However, OS domain adaptation is used only when the source and target data share a few classes of interest. Wei Zhang et al. introduced OS domain adaptation approach based on deep learning for addressing machinery fault diagnosis challenges. Their method leverages adversarial learning to extract domain-invariant features from both source and target domains. Additionally, it incorporates an instance-level weighted mechanism for the identification of outlier classes within the target domain [61]. Jinghui Tian et al. proposed an OS multi-source domain adaptation method based on deep learning, which incorporates a complementary transferability metric specifically designed for mechanical fault diagnosis. In this work, both the source and target domain include labelled data. The proposed method is tested on a mechanical fault diagnosis dataset, and the experimental results demonstrate that the method achieves higher accuracy in prediction of unknown fault types [62]. Xiaolei Yu et al. a statistical identification-guided OS domain adaptation method for fault diagnosis. They proposed the use of an Extreme Value Theory-guided Progressive Adaptation (EVTPA) approach to address limitations identified in prior OS fault diagnosis research. To evaluate their method, experiments were conducted using three datasets: the Office-31 dataset, the Planetary gearbox dataset, and the Wheelset bearing dataset. The experimental results indicate that the EVTPA model effectively identifies unknown-class samples and guides knowledge transfer for

Open-Set Domain Adaptation (ODA) tasks [63]. Huaqing Wang et al. proposed a novel approach to fault diagnosis using weighted domain adaptation with double classifiers. The authors demonstrate the effectiveness of their approach through experiments on several mechanical fault datasets: Laboratory Gearbox dataset, the Case Western Reserve University (CWRU) dataset, the Intelligent Maintenance Systems (IMS), the centrifugal pump dataset. They compare their method with other domain adaptation methods and show that it outperforms them in terms of accuracy and robustness [64]. Pau Panareda Busto et al. proposed the concept of open-set domain adaptation in the context of image classification and action recognition. They conducted evaluations on various combinations of source and target domains, including 26 open-set and 34 closed-set scenarios. The datasets used in their study encompassed the Office dataset and its extension with the Caltech dataset, the Cross-Dataset Analysis dataset, the Sentiment dataset, synthetic data, as well as two action recognition datasets-the Kinetics Human Action Video Dataset and the UCF101 Action Recognition Dataset [65]. Zhang et al. proposed a method that combines domain adaptation with an adaptive convolutional neural network for the purpose of visual inspection of steel surface defects [66]. Alvaro Fuentes et al. proposed an open-set self- and across-domain adaptation architecture for tomato disease recognition. The approach provides a set of guidelines for analyzing the recognition of novel data to make the system more adaptable to real-world environments. The proposed method is evaluated on a tomato plant diseases dataset, and the experimental results demonstrate that the proposed architecture achieved outstanding performance compared to other existing methods used for a similar task [67]. Fevziye Irem Eyiokur et al. proposed domain adaptation for unconstrained ear recognition using deep convolutional neural networks. They found that domain adaptation improves the performance of deep CNN models for ear recognition. Additionally, they observed that cropping ear images from profile faces and applying data augmentation enhance the accuracy of the ear recognition system [68]. Jieli Zhou et al. proposed the Semi-supervised Open-set Domain Adversarial network (SODA) for COVID-19 chest X-ray classification. The model addresses large domain shift and the limited scale of the COVID-19 chest X-ray dataset within a semi-supervised open-set domain adaptation framework. SODA achieves competitive results in closed-set domain adaptation and state-of-the-art performance in unsupervised and semi-supervised open-set domain adaptation [69]. Mohammed Abdelwahab and Carlos Busso investigated supervised domain adaptation for speech-based emotion recognition. They explored the impact of labeled data quantity, speaker diversity, and model adaptation approaches. Findings show even a small labeled dataset significantly improves performance, while increased speaker diversity doesn't yield substantial gains. Classifiers trained with naturalistic and spontaneous data perform similarly. Both domain adaptation and incremental training yield comparable results. Adaptive SVM and online SVM methods are proposed for model adaptation, with both showing similar performance [70]. Jun Zhang et al. proposed an open set domain adaptation network (OSDANet) in remote sensing image scene classification, addressing challenges with unknown classes. OSDANet uses an adversarially trained feature generator and classifier to determine alignment or rejection of target samples. Experimental results on AID and UC Merced datasets demonstrate the method's effectiveness, utilizing standard cross-entropy loss for categorizing source samples and binary cross-entropy loss for handling unknown samples during training [71]. Nirmal S. et al. proposed an efficient hyperspectral image (HSI) classification model incorporating open-set domain adaptation and Generative Adversarial Network (GAN) techniques. The model achieves high accuracy for known and unknown classes, simplifying annotation and feature classification across diverse HSI datasets [72]. In Table 1, we have presented the analysis of best studies following the OS domain adaptation.

II. MATHEMATICAL MODEL

Two different dynamical systems namely, a thermal pulse combustor and a nonlinear aeroelastic system (with three different types of nonlinearities) are considered for this study. The numerical models for these systems and subsystems are provided briefly in this section. The main aim of describing these numerical models is to show how differently these systems are modelled, and yet, contain similarities in the dynamical transitions.

A. PULSE COMBUSTOR

A pulse combustor comprises a combustion chamber, which is frequently modeled as a well-stirred reactor due to its vigorous mixing, and a tailpipe extending from one end of the combustor (as depicted in Fig. 1). Theoretical investigations of thermal pulse combustors employed a lumped model consisting of four coupled nonlinear ordinary differential equations. This model, which includes an unsteady well-stirred reactor and a lumped representation of a tailpipe, was originally developed by Richards et al. [73] and subsequently modified and explored by Mukhopadhyay and colleagues [10], [11], [21], [22]. These modifications incorporated considerations for radiative heat loss from the flames, introducing an additional layer of nonlinearity. The set of four coupled ordinary differential equations is derived from principles of mass conservation, fuel (species) mass

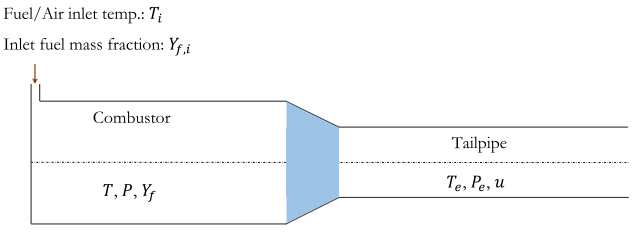

FIGURE 1. Schematic diagram of the Pulse combustor.

TABLE 1. Summary of reviewed papers on open set domain adaptation.

Author Name and Year	Model Used	Purpose	Data Set	Results
Wei Zhang, Xiang Li, Hui Ma, Zhong Luo, and Xu Li, 2021 [61]	An extension of DA methods, Instance-level Weighted Adversarial Learning	Machinery Fault Diagnosis	Case Western Reserve University (CWRU) and Train bogie	The proposed method demonstrated the ability to achieve high accuracy in cross-domain fault diagnosis testing and effectively identified the target outlier classes.
Jinghui Tian, Dongying Han, Hamid Reza Karimi, Yu Zhang, and Peiming Shi, 2023 [62]	An extension of DA methods, Open Set Multi-Source Domain Adaptation (OSMDA)	Machinery Fault Diagnosis	CWRU, Rolling mill (RM), and Gearbox dataset	The proposed method effectively distinguishes known fault classes and efficiently detects previously unidentified target classes, outperforming other domain adaptation techniques in various open-set fault diagnosis scenarios.
Xiaolei Yu, Zhibin Zhao, Xingwu Zhang, Xuefeng Chen, and Jianbing Cai, 2023 [63]	An extension of DA methods, utilizing Deep Neural Networks and guided by Extreme Value Theory (EVT), for progressive adaptation	Machinery Fault Diagnosis	Office-31 dataset, Planetary gearbox dataset, and Wheelset bearing dataset	The proposed method effectively identifies unknown-class samples, aids knowledge transfer in open-domain adaptation (ODA) tasks, outperforms comparable methods, and enhances industrial machinery safety and reliability.
Huaqing Wang, Zhitao Xu, Xingwei Tong, and Liuyang Song, 2023 [64]	An extension of DA methods, Weighted Domain Adaptation With Weighted Classifiers (WDADC)	Machinery Fault Diagnosis	Laboratory Gearbox dataset, CWRU, Intelligent Maintenance Systems (IMS), and Centrifugal pump dataset	The proposed method outperforms traditional domain adaptation techniques in open-set fault diagnosis. It achieves over 95% accuracy in various gearbox working conditions and surpasses other methods with an average accuracy of over 80% in equipment transfer tasks.
Pau Panareda Busto, Ahsan Iqbal, and Juergen Gall, 2020 [65]	Deep Neural Networks	Image and Action Recognition	Office dataset: Amazon, DSLR, and Webcam. Extension of the Office dataset with the Caltech dataset, the Cross-Dataset Analysis, the Sentiment dataset, synthetic data, and two action recognition datasets: the Kinetics Human Action Video Dataset and the UCF101 Action Recognition Dataset.	The study achieves state-of-the-art results for open-set domain adaptation and competitive results for closed-set domain adaptation in image classification and action recognition tasks.
Siyu Zhang, Qiuju Zhang, Jiefei Gu, Lei Su, Ke Li, and Michael Pecht, 2020 [66]	An extension of DA methods, Domain Adaptation-Adaptive Convolutional Neural Network (DA-ACNN)	Visual Inspection of Steel Surface Defects	The Northeast University (NEU) surface defect database	The experimental results show that DA-ACNN achieves higher accuracy than other widely employed methods.
Alvaro Fuentes, Sook Yoon, Taehyun Kim, and Dong Sun Park, 2021 [67]	Deep Neural Networks	Tomato Disease Recognition	Tomato plant diseases dataset	The proposed architecture achieved an outstanding mean average precision (mAP) of 94.76%, outperforming other existing methods used for similar tasks.
Fevziye Irem Eyiokur, Dogucan Yaman, Hazım Kemal Ekenel, 2017 [68]	Deep convolutional neural networks	Ear Recognition	Multi-PIE ear and the UERC data sets	The experimental results show that cropping ear images from profile faces and data augmentation enhance the accuracy of the ear recognition system.
Jieli Zhou, Baoyu Jing, Zeya Wang, Hongyi Xin, and Hanghang Tong, 2022 [69]	An extension of DA methods, Semi-supervised Open Set Domain Adversarial network (SODA)	COVID-19 Detection in Chest X-Rays	COVID-19 chest x-ray dataset	The proposed method shows that it achieves better AUC-ROC scores than the recent state-of-the-art models.
Mohammed Abdelwahab and Carlos Busso, 2015 [70]	Adaptive SVM algorithm and incremental SVM training method.	Emotion Recognition from Speech	IEMOCAP, SEMAINE, and RECOLA database.	The study observes a significant improvement in the performance of speech emotion recognition systems.
Jun Zhang, Jiao Liu, Lukui Shi, Bin Pan, and Xia Xu. 2020 [71]	The open set domain adaptation network (OSDANet)	Remote Sensing Image Scene Classification	The AID and UC Merced dataset	The experimental results demonstrate that the proposed method recognizes the known and unknown classes as accurately as possible.
S. Nirmal, V. Sowmya, K. P. Soman, 2020 [72]	Generative Adversarial Network	Hyperspectral Image Classification	Salinas and PaviaU dataset	The proposed model achieves high classification accuracy for unknown classes, with 99.07% accuracy for the Salinas dataset and 81.65% accuracy for the Pavia dataset.

conservation, energy conservation within the combustor, and momentum conservation within the tailpipe. These equations can be expressed as follows,

$$\frac{dT}{dt} = \frac{\gamma T}{P}\left(\frac{Z_i T_i}{\tau_f} + \frac{1}{\tau_c} + \frac{1}{\tau_h}\right) - \frac{T^2}{P}\left((\gamma-1)\frac{Z_e}{\rho_0} + \frac{Z_i}{\tau_f} + \frac{\gamma}{\tau_h T_w}\right) \tag{1}$$

$$\frac{dP}{dt} = \gamma\left(\frac{Z_i T_i}{\tau_f} + \frac{1}{\tau_{th}} + \frac{1}{\tau_c}\right) - \gamma T\left(\frac{Z_e}{\rho_0} + \frac{1}{\tau_h T_w}\right) \tag{2}$$

$$\frac{dY_f}{dt} = \frac{TZ_i}{P}\frac{1}{\tau_f}\left(Y_{f,i} - Y_f\right) - \frac{1}{\tau_c}\frac{C_p T_o T}{\Delta h_c P}\Bigg) \tag{3}$$

$$\frac{du}{dt} = (P_e - 1)\left(\frac{RT_0\tau_f}{L_{TP}L_{c,2}}\right) - \frac{1}{D_{TP}}\frac{L_{c,2}f}{2\tau_f}\frac{u^3}{|u|} \tag{4}$$

where, T represents the combustor temperature, T_w signifies the wall temperature, T_0 stands for the ambient temperature, P corresponds to the combustor pressure, C_p denotes the specific heat at constant pressure, τ_h represents the heat transfer time (Eq. 6), τ_c is the chemical reaction time (Eq. 7), τ_f is the flow time (Eq. 5), ρ_0 corresponds to the ambient density, γ signifies the ratio of specific heats, L_{TP} represents the length of the tailpipe, $L_{c,2}$ is the second characteristic length, u represents the gas velocity in the tailpipe, $y_{f,i}$ denotes the inlet fuel mass fraction, D_{TP} denotes the diameter of the tailpipe, y_f is the fuel mass fraction and f signifies the friction factor.

The equations presented above incorporate three fundamental characteristic times, namely, flow time (τ_f), heat transfer time (τ_h), and chemical reaction time (τ_c). These

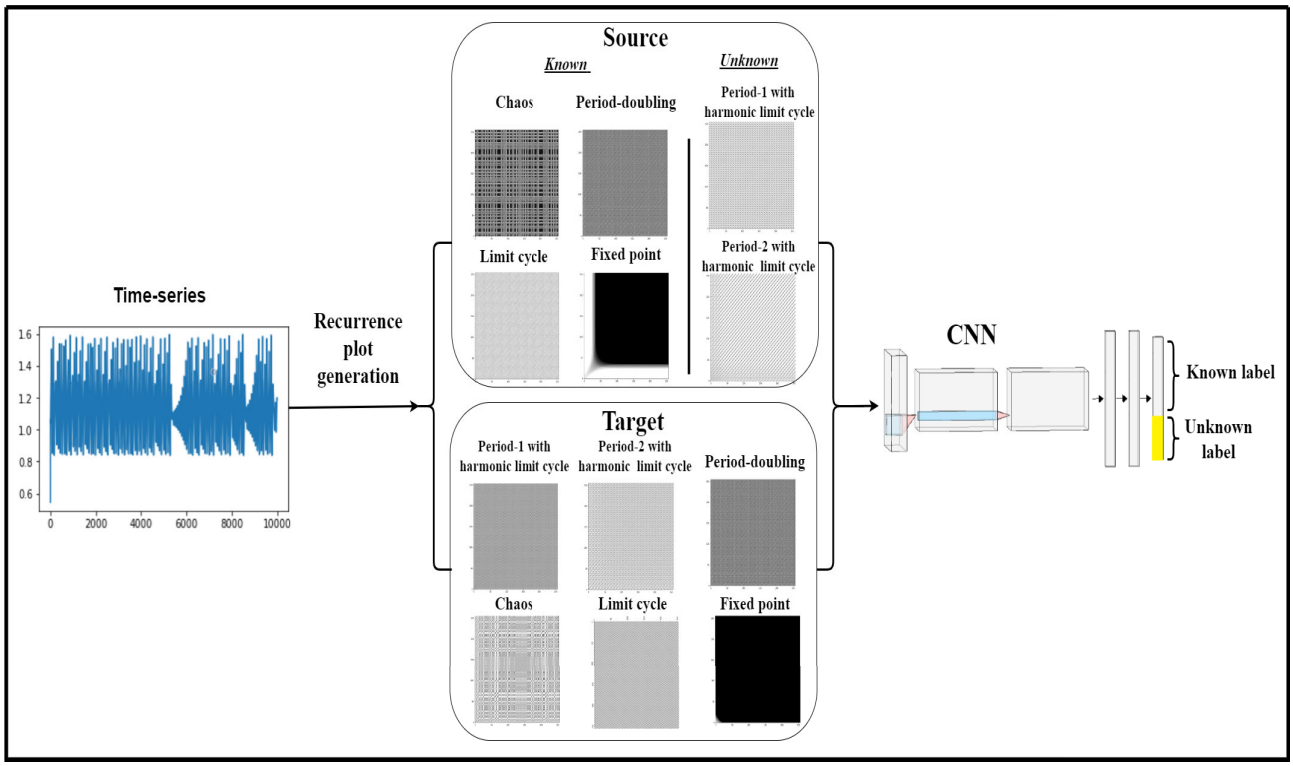

FIGURE 2. Pipeline of the proposed method: Convolutional neural network with open set domain adaption to classify the dynamical states of nonlinear fluid dynamical systems.

TABLE 2. Experiments done in this proposed work.

Cross-Domain Experiments	Source domain	Target domain		
1	Pulse combustor	Aeroelastic system with cubic nonlinearity	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with combined freeplay and aerodynamic nonlinearity
2	Aeroelastic system with freeplay nonlinearity	Pulse combustor	Aeroelastic system with cubic nonlinearity	Aeroelastic system with combined freeplay and aerodynamic nonlinearity
3	Aeroelastic system with combined freeplay and aerodynamic nonlinearity	Pulse combustor	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with cubic nonlinearity
4	Aeroelastic system with cubic nonlinearity	Pulse combustor	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with combined freeplay and aerodynamic nonlinearity

characteristic times are defined as follows:

$$\tau_f = \frac{Z_i}{\rho_0}, \tag{5}$$

$$\tau_h = \frac{L_{c,1}\rho_0 C_p T_0}{h_{eff} T_w}, \tag{6}$$

$$\tau_c = \left[\frac{B\Delta h_c P^2 {y_f}^2}{C_p T_0 T^{1.5}} e^{-\frac{T_a}{T}}\right]^{-1} \tag{7}$$

where $L_{c,1}$ represents the initial characteristic length, V corresponds to the combustor's volume, $Z_i = \dot{m}_i/V$, $\dot{m}_i$ denotes the mass flow rate at the combustor inlet, B is the pre-exponential factor, T_a denotes the activation temperature and h_{eff} represents the effective heat transfer coefficient.

The effective heat transfer coefficient, encompassing contributions from both convection and radiation, is defined as

$$h_{eff} = h + 4k_p\sigma L_{c,1} {T_0}^3 \left({T_w}^2 + T^2\right)\left(T_w + T\right), \tag{8}$$

where, h represents the convective heat transfer coefficient, k_P stands for the Planck mean absorption coefficient, and σ corresponds to the Stefan-Boltzmann coefficient. Z_e $(= \dot{m}_e/V)$ is derived from the conservation of mass within the tailpipe as

$$Z_e = \frac{uP_e}{\tau_f T_e}, \tag{9}$$

where, P_e denotes the pressure within the tailpipe, and T_e represents the temperature within the tailpipe. The flow through the nozzle that connects the combustor and the tailpipe is assumed to be isentropic, although irreversibilities are present in both the combustor and the tailpipe. As a result

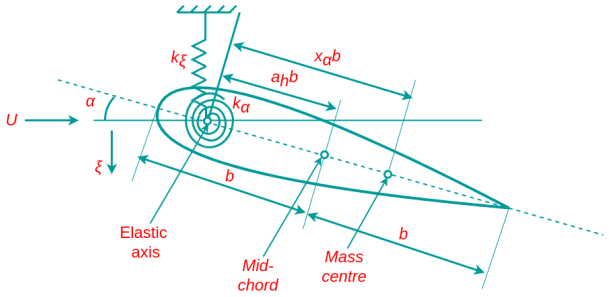

FIGURE 3. Schematic diagram of the 2-DoF aeroelastic model.

of these considerations, the temperature and pressure within the tailpipe are obtained as

$$T_e = T - \frac{u^2 {L_{c,2}}^2}{2C_p T_0 {\tau_f}^2}, \tag{10}$$

$$P_e = P\left(\frac{T_e}{T}\right)^{\gamma/(\gamma-1)}, \tag{11}$$

The detailed derivation of the model can be obtained in Datta et al. [11]

B. AEROELASTIC SYSTEM

The aeroelastic system under consideration is shown in Fig. 3. The system consists of a 2-DoF NACA 0012 airfoil of semi-chord length b, having pitch (α) plunge (ξ) motions, constrained through translational and torsional springs of stiffness k_ξ and k_α, respectively. Terms, x_α and a_h denote the non-dimensionalized length of the mass center and mid-chord from the elastic axis, respectively. The equation of motion for the system is given as:

$$\xi'' + x_\alpha \alpha'' + \left(\frac{\bar{\omega}}{U}\right)^2 \xi = -\frac{1}{\pi\mu} C_l(\tau), \tag{12}$$

$$\frac{x_\alpha}{{r_\alpha}^2}\xi'' + \alpha'' + \left(\frac{1}{U}\right)^2 M(\alpha) = \frac{2}{\pi\mu {r_\alpha}^2} C_m(\tau). \tag{13}$$

here, $\bar{\omega}$ denotes the plunge to pitch natural frequency ratio; μ denotes the nondimensional mass ratio; r_α is the non-dimensional radius of gyration; U is the non-dimensional flow speed; C_l and C_m represent the aerodynamic lift and moment coefficients, respectively. $M(\alpha)$ denotes the non-dimensional pitch stiffness and is modelled as a cubic polynomial (for cubic hardening nonlinearity) or a bilinear function (for freeplay nonlinearity) to incorporate structural nonlinearity. The dynamics of the aeroelastic system is analysed under three different cases of nonlinearities. Details of the respective mathematical models are given as follows:

1) AEROELASTIC SYSTEM WITH CUBIC NONLINEARITY

Structural nonlinearity that arises from large deformations is generally modelled via Cubic nonlinear springs. Accordingly, $M(\alpha)$ in Eq. 13 is defined as-

$$M(\alpha) = \alpha + \beta_\alpha \alpha^3 \tag{14}$$

where β_α is the coefficient of cubic stiffness. The aerodynamic lift and moment coefficients are calculated using Wagner's unsteady attached flow model. More details of the model are not provided here for the sake of brevity and can be found in [29].

2) AEROELASTIC SYSTEM WITH FREEPLAY NONLINEARITY

Loose hinges or worn parts in aeroelastic systems are modelled as freeplay nonlinearity. Here, pitch stiffness is considered to have a freeplay nonlinearity and is modelled as a bilinear function-

$$M(\alpha) = \begin{cases} \alpha + \delta & if \quad \alpha < -\delta \\ 0 & if \quad -\delta \le \alpha \le \delta \\ \alpha - \delta & if \quad \alpha > \delta \end{cases} \tag{15}$$

here δ denotes the freeplay gap and is assumed to be 0.5 in this paper. The unsteady aerodynamic loads are calculated using Wagner's function.

3) AEROELASTIC SYSTEM WITH COMBINED FREEPLAY AND AERODYNAMIC NONLINEARITY

In this case, both structure and aerodynamic loads are modelled as nonlinear. The structure possesses a pitch freeplay nonlinearity, same as II-B2, and the aerodynamics is governed by dynamic stall nonlinearity [3], [26]. The equation of motion is given by Eqs. 12 and 13. The aerodynamic loads are modelled using a semi-empirical Leishman-Beddoes (LB) dynamic stall model [23], [29]. The LB model is capable of capturing the different stages of dynamic stall *i.e.* flow separation, vortex shedding, and flow reattachment.

The LB model can be represented as-

$$x' = f(x, \hat{\alpha}, q), \tag{16}$$

states $x = [x_1, x_2, \ldots, x_{12}]^T$ corresponds to aerodynamic loads during different dynamic stall regimes; $q = 2\alpha'$ is the nondimensional effective pitch rate, and $\hat{\alpha}$ is the effective angle of incidence.

The aerodynamic load coefficients are given as

$$\begin{Bmatrix} C_n \\ C_m \\ C_c \end{Bmatrix} = g(x, \alpha, q), \tag{17}$$

where C_c and C_n are the coefficients of chord force and the normal force, respectively. The lift coefficient is given as,

$$C_l = C_n \cos\alpha - C_c \sin\alpha. \tag{18}$$

full details of the model are not provided here for the sake of brevity and can be found in [74].

III. METHODS

This section describes the datasets used in this work in depth. Second, we review the core ideas underlying the CNN model and recurrence plot generation. To assess the performance of the proposed CNN model, we discuss it along with the parameter choices and evaluation metrics.

TABLE 3. Dynamical states in pulse combustor dataset.

Dataset	Dynamical states	Number of time-series samples
Pulse combustor	Chaos	67
	Period-doubling	54
	LCO	126
	Fixed point	34

A. DATASET DESCRIPTION

In this study, we consider the pressure time series of the pulse combustor model and the pitch angle time series of the aeroelastic model. The pulse combustor responses are obtained by solving the Eqs. 1, 2, 3, and 4, and aeroelastic responses are obtained by solving the Eqs. 12 and 13 in state space form using the fourth-order Runge-Kutta numerical time integration technique. For the pulse combustor, the time-step is 1×10^{-5} s. Thus the dataset generated over the time interval 2.5 s, is sufficiently large for post-processing analysis. The pressure time series of the pulse combustor model consists of 2,00,000 data samples (removing the initial transience), whereas, the pitch angle time series of the aeroelastic model consists of 80,000 data samples. To reduce the computational cost of generating the recurrence plot, we down-sample the time series sequences of the pulse combustor model and aeroelastic model with $1/20^{th}$ and $1/8^{th}$ of the original sampling frequency. As a result, the resampled times series only has 10,000 data samples.

1) PULSE COMBUSTOR

Figure 4 shows the bifurcation diagram where the peak values of temperature oscillations are plotted with the bifurcation parameter, here, wall temperature. The pulse combustor model considered in this work exhibits four dynamical states, namely, (i) chaotic oscillations, (ii) period-doubling oscillations, (iii) limit cycle oscillations, and (iv) fixed-point response, as the wall temperature is increased (see Fig. 4). The dataset consists of 280 sequences of pressure signal samples at different wall temperatures. Among 280 sequences of pressure signals, 67 sequences are annotated as chaotic, 54 sequences are annotated as period-doubling bifurcation, 126 sequences are annotated as LCO, and 34 sequences are annotated as a fixed point (Table 3).

2) AEROELASTIC SYSTEM WITH CUBIC NONLINEARITY

Figure 5 shows the bifurcation diagram of the model. The considered model has two dynamical states (Table 4). Among 81 sequences of pitch angle measurements, 26 sequences are annotated as fixed points, and 55 sequences are annotated as LCO.

3) AEROELASTIC SYSTEM WITH FREEPLAY NONLINEARITY

Next, the pitch-plunge aeroelastic response data is obtained from a nonlinear aeroelastic structure with structural freeplay nonlinearity. Figure 6 shows the bifurcation diagram of

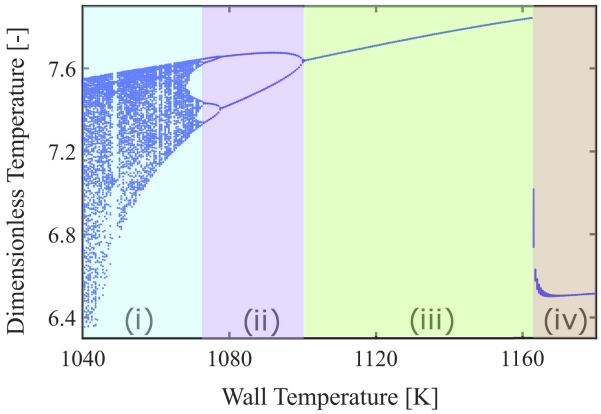

FIGURE 4. Bifurcation diagram (T vs T_W) of pulse combustor model. The regimes for different dynamical states are shown as (i) chaotic oscillations, (ii) period-doubling oscillations, (iii) limit cycle oscillations, and (iv) fixed-point response.

TABLE 4. Dynamical states in aeroelastic system with cubic nonlinearity dataset.

Dataset	Dynamical states	Number of time-series samples
Aeroelastic system with cubic nonlinearity	LCO	55
	Fixed point	26

TABLE 5. Dynamical states in aeroelastic system with freeplay nonlinearity dataset.

Dataset	Dynamical states	Number of time-series samples
Aeroelastic system with freeplay nonlinearity	LCO	22
	Fixed point	8
	Chaos	7
	p-1-h LCO	12
	p-2-h LCO	11

the model. The details of the model and the bifurcation diagram can be found in [23]. The considered model has 5 dynamical states (Table 5). Among 60 sequences of pitch angle measurements, 8 sequences are annotated as a fixed point, 22 sequences are annotated as LCO, 7 sequences are annotated as chaotic, 12 sequences are annotated as period-1 with harmonic (p-1-h) LCO, and 11 sequences are annotated as period-2 with harmonic (p-2-h) LCO.

4) AEROELASTIC SYSTEM WITH COMBINED FREEPLAY AND AERODYNAMIC NONLINEARITY

This study used data from the model of aeroelastic response with aerodynamic nonlinearity. Figure 7 shows the bifurcation diagram of the model. The details of the model and the bifurcation diagram can be found in [23]. The model considered has 4 dynamical states (Table 6). Among 79 sequences of pitch angle measurements, 10 sequences are annotated as a fixed point, 7 sequences are annotated as chaotic, 48 sequences are annotated as p-1-h LCO, and 14 sequences are annotated as p-2-h LCO.

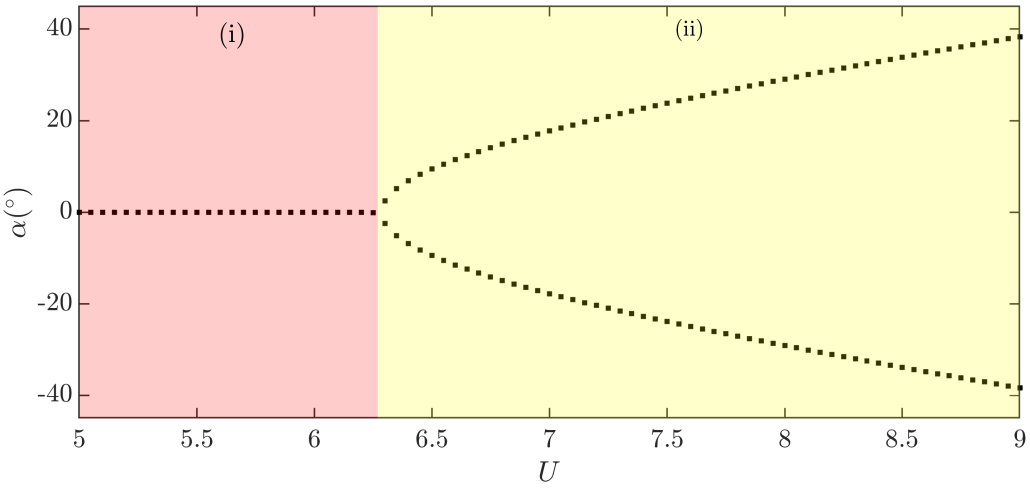

FIGURE 5. Bifurcation diagram (α vs U) for an aeroelastic system with a structural cubic hardening nonlinearity. Regime (I) denotes the fixed point response and regime (ii) denotes the LCO response.

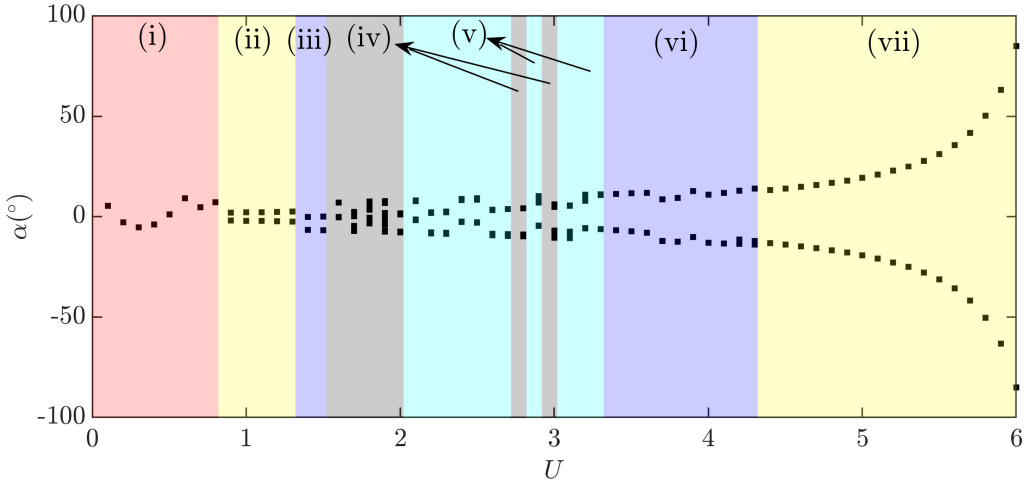

FIGURE 6. Bifurcation diagram (α vs U) for an aeroelastic system with combined structural freeplay and aerodynamic nonlinearity. Regime (i) denotes the fixed point response, regime (ii) denotes the LCO response, regime (iii) denotes the p-1-h LCO, regime (iv) denotes the chaotic response, regime (v) denotes the p-2-h LCO response, regime (vi) denotes the p-1-h LCO and regime (vii) denotes the LCO response.

TABLE 6. Dynamical states in aeroelastic response with combined freeplay and aerodynamic nonlinearity dataset.

Dataset	Dynamical states	Number of time-series samples
Aeroelastic system with combined freeplay and aerodynamic nonlinearity	Aperiodic LCO	7
	LCO	48
	p-1-h LCO	14
	Fixed point	10

B. RECURRENCE PLOT GENERATION:

Recurrence is a fundamental characteristic of deterministic dynamical systems, and recurrence plots provide a visual tool for identifying the time instances when the system's trajectory nearly revisits the same region within the phase space [75]. The primary goal is to identify the instances when particular trajectories revisit a previous state, and it can be expressed as follows:

$$R_{i,j} = \Theta\left(\varepsilon - \left\| \overrightarrow{x_i} - \overrightarrow{x_j} \right\|\right), \overrightarrow{x}(.) \in \Re^m, i,j = 1, \ldots, K \quad (19)$$

In Eq. 19, $\mathbf{x}$ represents the phase space vector, reconstructed from the time series applying Takens' delay embedding theory, Θ is the Heaviside function, ε represents the threshold distance, $\|.\|$ represents the Euclidean norm, and the indices represent the instants of time. The matrix $R_{i,j}$ is a symmetric matrix consisting of ones and zeros, and it can be visualized in a 2-dimensional format as a recurrence plot to depict how trajectories evolve over time. In the recurrence plot, 'ones' are denoted by black points, signifying time instances where the pairwise distances fall below the threshold ε, while 'zeroes' are represented by white points, indicating time instances where the pairwise distances exceed ε.

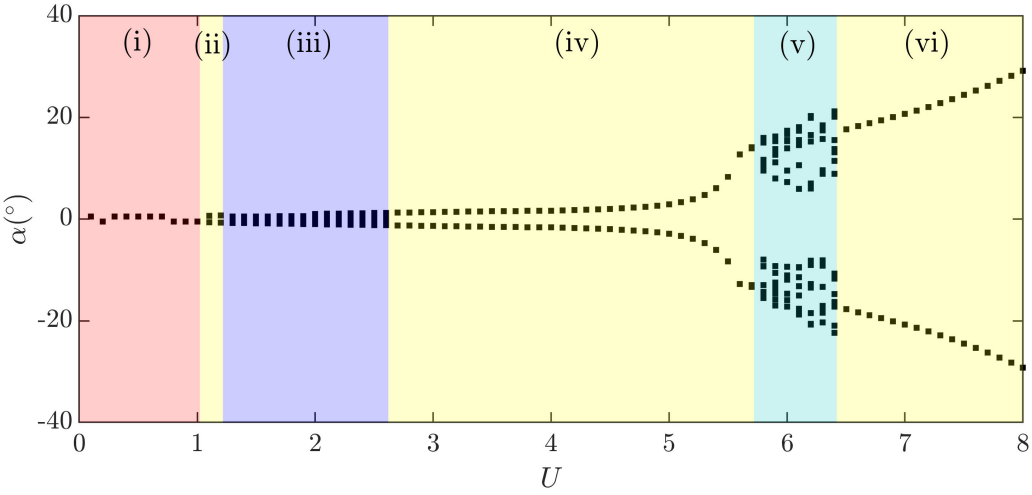

FIGURE 7. Bifurcation diagram (α vs U) for an aeroelastic system with combined structural freeplay and aerodynamic nonlinearity. Regime (i) denotes the fixed point response and regime (ii) denotes the LCO response, regime (iii) denotes the P-1-h LCO, regime (iv) denotes the LCO response, regime (v) denotes the aperiodic LCO response and regime (vi) denotes the LCO response.

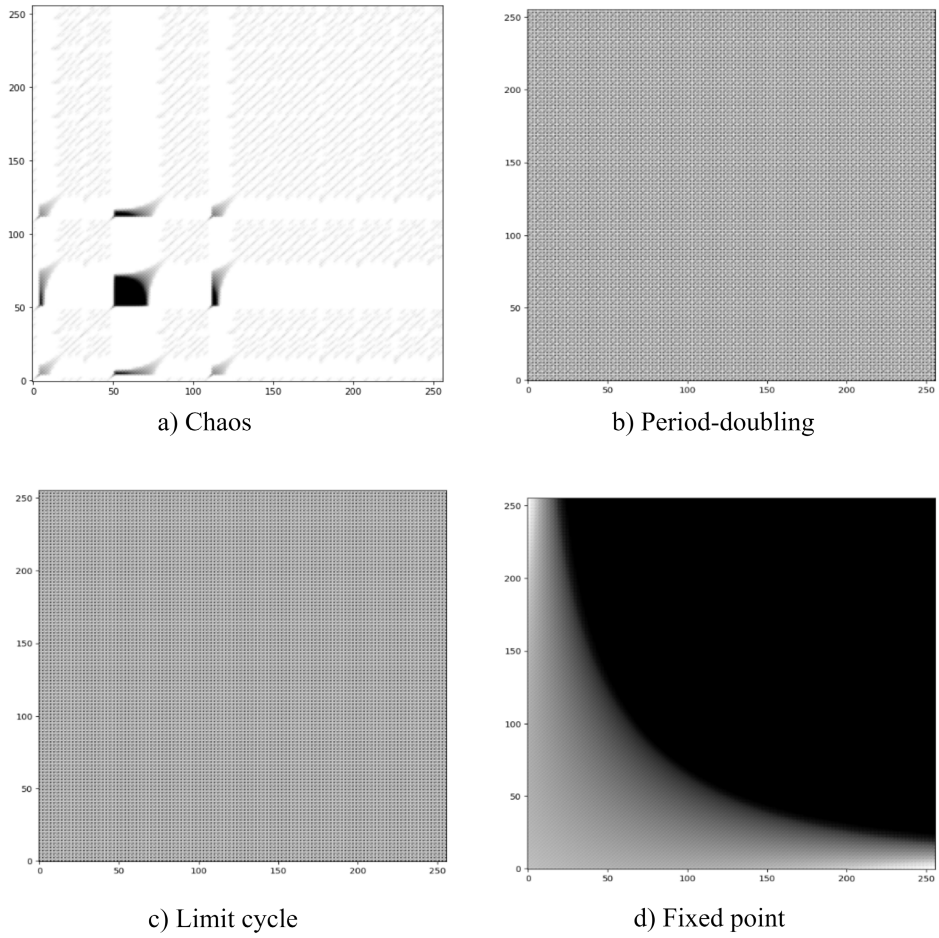

FIGURE 8. Sample of the recurrence plot representation of the time series dataset.

The first step in generating a recurrence plot is to reconstruct the phase space from the time series data, requiring two fundamental parameters: time delay and embedding dimension. The choice of appropriate parameter values is

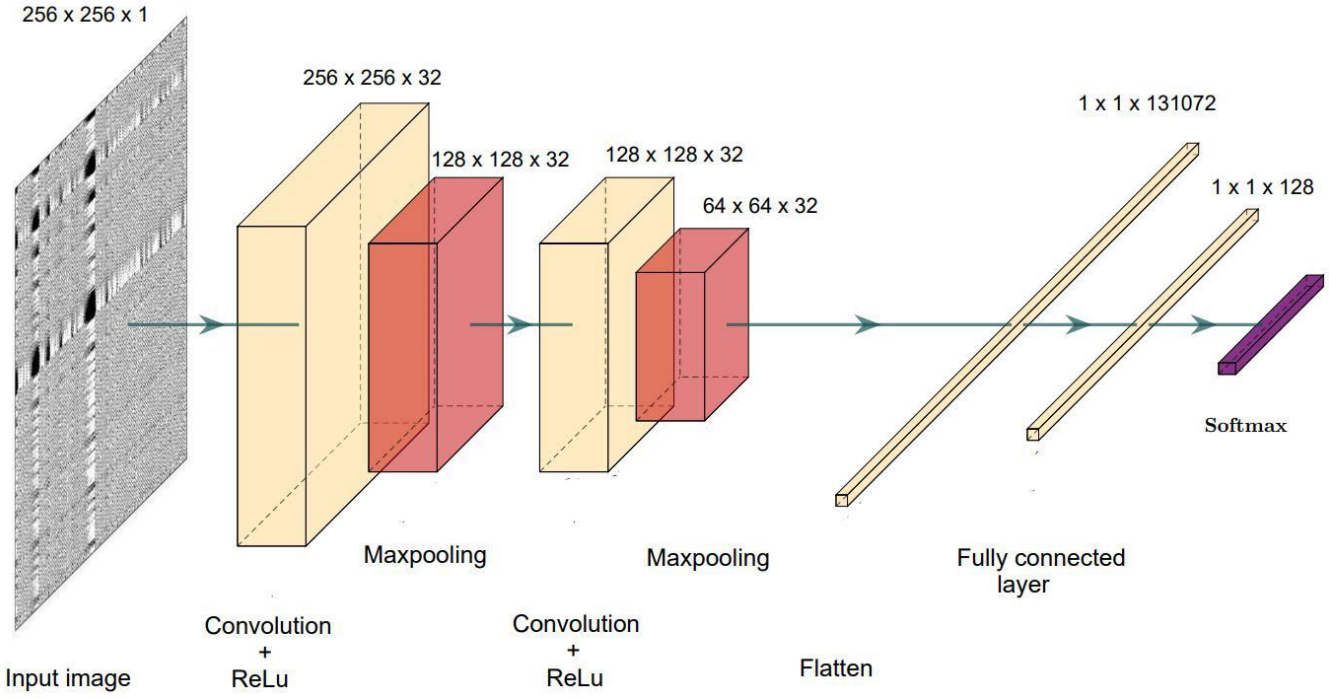

FIGURE 9. Schematic diagram of the CNN model used in this study.

TABLE 7. CNN model architecture.

Layers	Type	Subtype	Configuration	Activation
1	Input Layer	Image Input	256 x 256 x 1 images	256 x 256 x 1
2	Feature Extraction Layer	Convolution	32 convolutions: 3x3x1 filters, stride 1, 'same' padding	256 x 256 x 32
3		ReLU	ReLu	256 x 256 x 32
4		Max Pooling	2x2 max-pooling: stride 2, no padding	128 x 128 x 32
5		Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	128 x 128 x 32
6		Relu	ReLu	128 x 128 x 32
7		Max Pooling	2x2 max-pooling: stride 2, no padding	64 x 64 x 32
8	Classification Layer	Dropout	50% dropout	64 x 64 x 32
9		Fully Connected	1 dense layer with 128 neurons	128
10		Softmax	softmax	
11	Output Layer	Classification Output	No of classes	

crucial as they directly impact classification outcomes. If the embedding dimension is too low, it may lead to false recurrence, while choosing a dimension that is too large can result in round-off errors [76]. The time lag value should be selected to ensure that the elements of the embedding vectors are no longer correlated, enabling the accurate extraction of spatial or geometrical structures for subsequent analysis. In this study, the time lag value was determined using the first local minimum of the average mutual information [77], and the value of the embedding dimension was determined using the false closest neighbour algorithm [78]. In this study, we set the threshold value ϵ to 10. Recurrence plots were generated using the recurrencePlot function in MATLAB. Additionally, we applied the bicubic interpolation algorithm to resize the recurrence plots to 256 $\times$ 256, reducing the computational load during the training of the proposed CNN model. A sample of these generated recurrence plots is displayed in the figure 8.

C. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNNs) are among the most widely used deep neural network (DNN) techniques. LeCun et al. [44] was the first to apply CNN to recognize hand-written digits. Then, the method was successfully applied to a myriad of application domains, especially in computer vision. In contrast to the conventional ''feature-based'' categorisation framework, CNN does not need manually created features. The feature learning and classification components are combined and cooperatively learned in a single model. As a result, the technique automatically derives features from raw input data [79]. However, CNN's architecture considerably influences how well the CNN

models function. The functioning of the CNN model depends upon the chosen CNN architecture. A typical architecture of CNN consists of repetitions of a stack of many convolution layers and a pooling layer, followed by one or more fully connected layers. The following sections provide more information on these layers.

1) PROPOSED CONVOLUTIONAL NEURAL NETWORK MODEL PARAMETERS

This study employs a CNN model with two feature extraction layers (our extensive experiments with the dataset revealed that increasing the depth of the CNN models did not result in a significant improvement in classification accuracy (see Section IV-E). Our results showed that the shallower model, as depicted in (Figure 9), achieved competitive and satisfactory performance). The feature extraction layers comprise a convolutional layer, a ReLU activation function, and max-pooling layers. The convolutional layer utilizes a 3 $\times$ 3 x 1 filter with a stride of [1, 1] and padding set to 'same,' resulting in a feature map size of 256 $\times$ 256 x 32. Following the application of the ReLU activation function, a 2 $\times$ 2 max-pooling operation with a stride of [1, 1] is applied to the feature map. Furthermore, to mitigate overfitting, we have employed regularization techniques such as L2 regularization, 50% dropout, and early stopping. In the proposed CNN model, a fully connected layer with 128 nodes was employed for classification, with the final classification carried out in the softmax layer. Additionally, a summary of the architectural details of the proposed CNN model is presented in Table7.

IV. RESULTS & DISCUSSIONS

In this section, we evaluated the performance of the proposed CNN-based open-set domain adaptation model for classifying dynamical states of nonlinear fluid dynamical systems through four cross-domain experiments. For all cross-domain experiments in this study, we split the dataset as follows: 40% for training, 30% for validation, and the remaining 30% for testing. The performance of the trained model was evaluated using 5-fold cross-validation.

Additionally, we conducted further experiments to address specific questions:

- What were the effects of increasing the number of feature extraction layers in the CNN model on classification performance?
- What is the effect of classifying using Short-Time Fourier Transform (STFT) images compared to using Recurrence Plots (RPs) as input for the proposed CNN model
- We also examined the interpretability of the CNN model by employing feature visualization techniques like GRAD-CAM. This allowed us to highlight significant features within the RPs of the time series related to the dynamical states contributing most to accurate predictions.
- Can traditional Machine Learning (SVM) achieve similar results to the proposed CNN model?

The experiments were conducted on a computer equipped with an AMD Ryzen 9 5900HX processor running at a clock speed of 3.30 GHz, a 6 GB GeForce RTX 3060 Graphics card, and 16 GB of RAM. MATLAB was employed for dataset preparation, while the Keras deep learning module was utilized to conduct the experiments.

A. EXPERIMENT 1: CROSS-DOMAIN CLASSIFICATION OF DYNAMICAL STATES USING CNN MODEL TRAINED ON PULSE COMBUSTOR DATASET.

In this experiment, we train the CNN with the data from the pulse combustor and employ it to detect dynamical states pertinent to other datasets (Table 2). The former is the source domain, and the latter is the target domain. This experiment's source and target domains share three dynamical states: chaos, fixed point, and LCO. On the other hand, the dynamical states p-1-h LCO, p-2-h LCO, and the aperiodic LCO are not present in the source domain. These dynamical states which are not present in the source domain are annotated as unknowns and trained along with source domain dynamical states. Table 8 shows the familiar and unknown dynamical states in experiment 1. During the training phase, a batch size of 10 was considered, and the model was trained for 200 epochs with the inclusion of an early stopping mechanism. The model achieves 99% training accuracy, and an accuracy of 96% is achieved in the validation result. The training and loss plots are shown in Figure 12 and 11. It is observed that there is minimal variation between train and test curves. This ensures that our proposed model did not overfit. Figure 10 presents the confusion matrix. It is evident from the confusion matrix that the CNN model achieves a test accuracy of 98%. In K-fold cross validation the model achieves an average test accuracy of 97.00 (+- 2.75) for all folds. The proposed CNN model achieves the highest accuracy of 100% in classifying the dynamical states of chaos, period-doubling bifurcation, fixed point, and unknown. The model also shows accuracy of 97% in identifying the LCO. It has been observed in the experiment that two images of LCO have been misclassified as unknown. Table 9 shows the proposed CNN model performance metric for cross-domain experiment 1.

B. EXPERIMENT 2: CROSS-DOMAIN CLASSIFICATION OF DYNAMICAL STATES USING CNN MODEL TRAINED ON AEROELASTIC SYSTEM WITH FREEPLAY NONLINEARITY DATASET.

In this experimental setup, we utilize the CNN model trained on data from the aeroelastic system with freeplay nonlinearity (the source domain) to identify relevant dynamical states in other datasets (Table 2). This experiment's source and target domains share four dynamical states: chaos, fixed point, p-1-h LCO, and LCO. On the other hand, The dynamic states period-doubling and aperiodic LCO are not present in the

TABLE 8. Common and unknown dynamical states in cross-domain experiment 1.

Source domain	Target domain			Common dynamical states in source & target domain	Unknown dynamical states in source domain
Pulse combustor	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with combined freeplay and aerodynamic nonlinearity	Aeroelastic system with cubic nonlinearity		
Chaos	Chaos	Aperiodic LCO	LCO	Chaos	Aperiodic LCO
Period-doubling	p-1-h LCO	LCO	Fixed point	LCO	p-1-h LCO
LCO	p-2-h LCO	p-1-h LCO		Fixed point	p-2-h LCO
Fixed point	LCO	Fixed point			
	Fixed point				

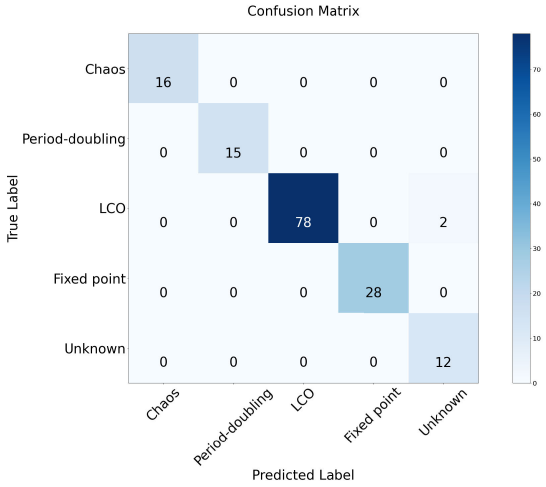

FIGURE 10. Confusion matrix for cross-domain experiment 1.

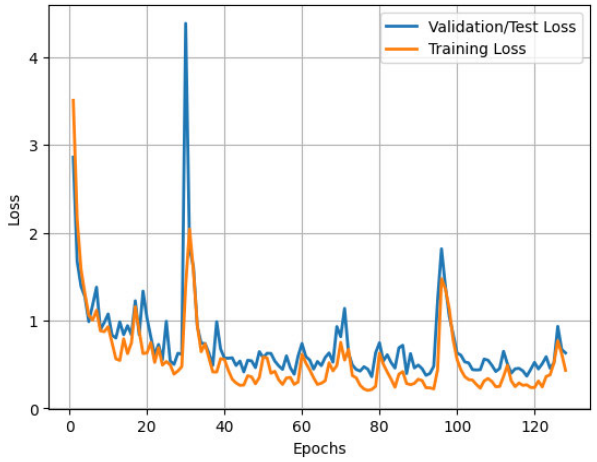

FIGURE 11. Training loss plot of cross-domain experiment 1.

TABLE 9. Performance metric of the proposed CNN model for cross-domain experiment 1.

Dynamical States	Prescision	Recall	F1-Score
Chaos	1.00	1.00	1.00
Period-doubling	1.00	1.00	1.00
LCO	1.00	0.97	0.99
Fixed point	1.00	1.00	1.00
Unknown	0.86	1.00	0.92

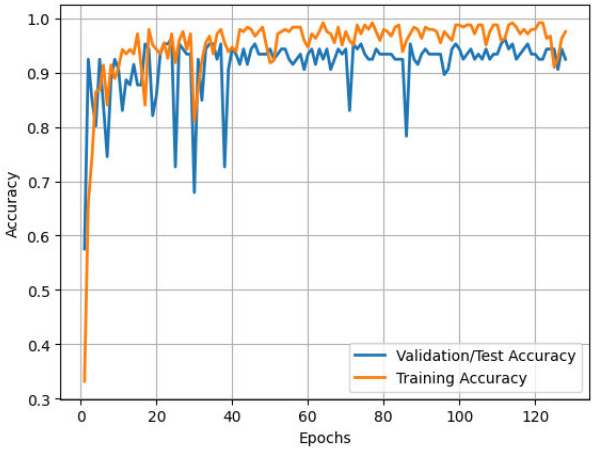

FIGURE 12. Training accuracy plot of cross-domain experiment 1.

source domain. These dynamical states were annotated as unknown states and trained along with subsets of dynamical states in the source domain. Table 10 shows the familiar and unknown dynamical states in experiment 2. During training, a batch size of 10 was considered, and the model was trained for 250 epochs. The model achieves 99% training accuracy, and an accuracy of 98% is achieved in the verification result. The training accuracy and loss plots are shown in Figure 15 and 14. It has been observed that the train and test curves differ only slightly. This guarantees that the suggested model was not overfit. The confusion matrix represented in Figure 13 reveals that the proposed CNN model achieves a test accuracy of 97%. In K-fold cross validation the model achieves an average test accuracy of 94.40 (+- 5.53) for all folds. In recognising the dynamical states of chaos LCO, and fixed point the proposedÂ CNN obtains the highest accuracy of 100%. Additionally, the algorithm has a 91% accuracy rate when identifying p-1-h LCO and a 81% accuracy rate for identifying unknown dynamical states. Furthermore, the model identifies the dynamical state p-2-h LCO with 50% accuracy. In this experiment, it is observed that the model misclassifies three images belonging to p-1-h LCO as LCO and one image of p-2-h LCO as p-1-h LCO and 3 images of unknown dynamical state as p-2-h LCO. The proposed CNN model's performance metric for cross-domain experiment 2 is shown in Table 11.

TABLE 10. Common and unknown dynamical states in cross-domain experiment 2.

Source domain	Target domain			Common dynamical states in source & target domain	Unknown dynamical states in source domain
Aeroelastic system with freeplay nonlinearity	Pulse combustor	Aeroelastic system with combined freeplay and aerodynamic nonlinearity	Aeroelastic system with cubic nonlinearity		
Chaos	Chaos	Aperiodic LCO	LCO	Chaos	Aperiodic LCO
p-1-h LCO	Period-doubling	LCO	Fixed point	LCO	Period-doubling
p-2-h LCO	LCO	p-1-h LCO		Fixed point	
LCO	Fixed point	Fixed point		p-1-h LCO	
Fixed point					

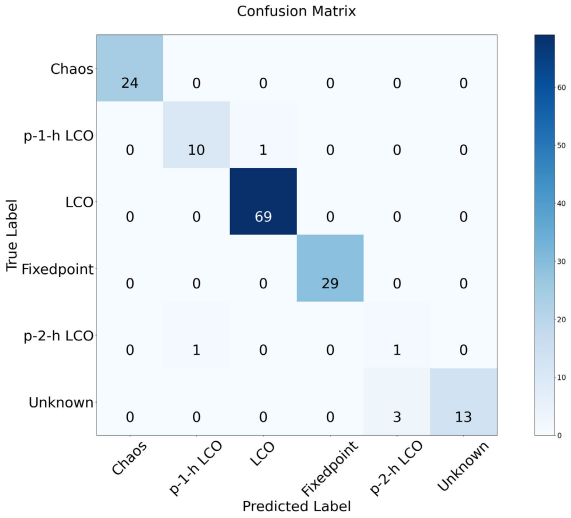

FIGURE 13. Confusion matrix for cross-domain experiment 2.

TABLE 11. Performance metric of the proposed CNN model for cross-domain experiment 2.

Dynamical States	Prescision	Recall	F1-Score
Chaos	1.00	1.00	1.00
p-1-h LCO	0.91	0.91	0.91
LCO	0.99	1.00	0.99
Fixed point	1.00	1.00	1.00
p-2-h LCO	0.25	0.50	0.33
Unknown	1.00	0.81	0.90

C. EXPERIMENT 3: CROSS-DOMAIN CLASSIFICATION OF DYNAMICAL STATES USING CNN MODEL TRAINED ON AEROELASTIC SYSTEM WITH COMBINED FREEPLAY AND AERODYNAMIC NONLINEARITY DATASET.

In this experimental setup, we train the CNN model using data from the aeroelastic system featuring both freeplay and aerodynamic nonlinearity (source domain), and then we apply it to identify relevant dynamical states in other datasets. (Table 2). This experiment's source and target domains share three dynamical states: fixed point, period-1 with LCO, and LCO. On the other hand, The dynamical states chaos, period-doubling, and p-2-h LCO are not present in the source domain. These dynamical states were labelled as unknown and trained alongside subsets of dynamical states present in the source domain. Table 12 shows the familiar and unknown dynamical states in experiment 3. During training, a batch

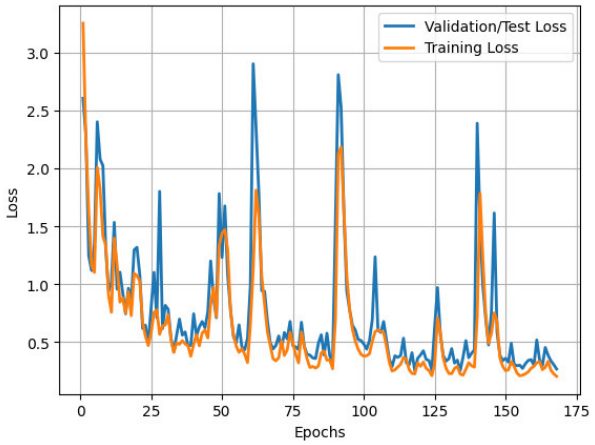

FIGURE 14. Training loss plot of cross-domain experiment 2.

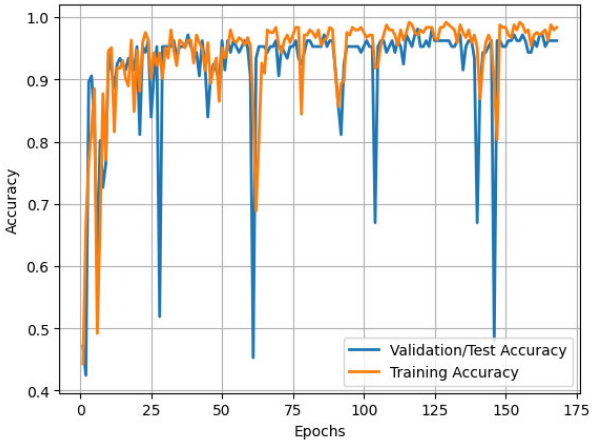

FIGURE 15. Training accuracy plot of cross-domain experiment 2.

size of 10 was considered, and the model was trained for 175 epochs. The model achieves 98% training accuracy, and an accuracy of 96% is achieved in the verification result. The training accuracy and loss plots are shown in Figure 17 and 18. Its evident from the plots that the model did not overfit as the variations in the training loss and test loss is similar. The confusion matrix in Figure 16 reveals that the proposed CNN model achieves a test accuracy of 96%. In K-fold cross validation the model achieves an average test accuracy of 95.80 (+- 1.47) for all folds. As can be seen from the confusion matrix, CNN achieves the highest accuracy

TABLE 12. Common and unknown dynamical states in cross-domain experiment 3.

Source domain	Target domain			Common dynamical states in source & target domain	Unknown dynamical states in source domain
Aeroelastic system with combined freeplay and aerodynamic nonlinearity	Pulse combustor	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with cubic nonlinearity		
Aperiodic LCO	Chaos	Chaos	LCO	p-1-h LCO	Chaos
LCO	Period-doubling	p-1-h LCO	Fixed point	LCO	Period-doubling
p-1-h LCO	LCO	p-2-h LCO		Fixed point	p-2-h LCO
Fixed point	Fixed point	LCO			
		Fixed point			

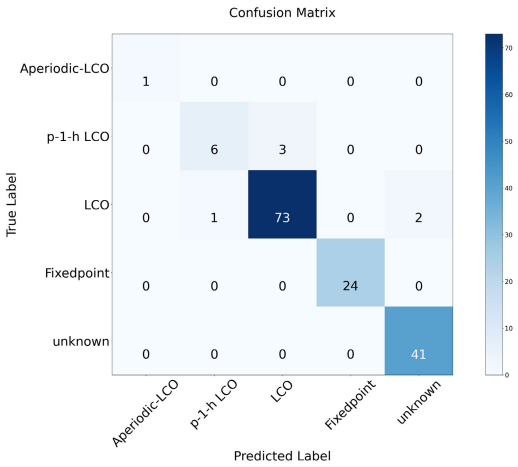

FIGURE 16. Confusion matrix for cross-domain experiment 3.

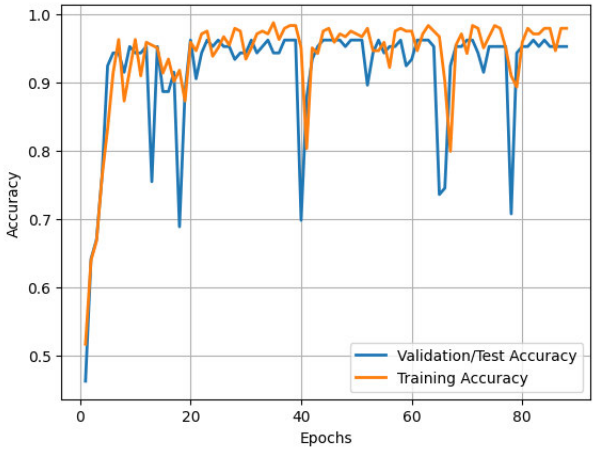

FIGURE 17. Training accuracy plot of cross-domain experiment 3.

TABLE 13. Performance metric of the proposed CNN model for cross-domain experiment 3.

Dynamical States	Prescision	Recall	F1-Score
Aperiodic LCO	1.00	1.00	1.00
p-1-h LCO	0.86	0.67	0.75
LCO	0.96	0.96	0.96
Fixed point	1.00	1.00	1.00
Unknown	0.95	1.00	0.98

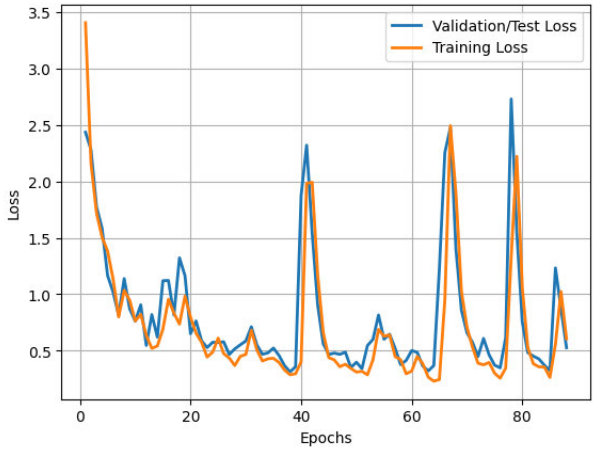

FIGURE 18. Training loss plot of cross-domain experiment 3.

of 100% in identifying dynamical states of Aperiodic LCO, fixed point, and Unkown. Furthermore, The model also shows high accuracy 96% in identifying LCO and accuracy of 67% in identifying p-1-h LCO. In this experiment, it is observed that the model misclassifies three images belonging to p-1-h LCO as LCO. In images belonging to LCO, one image is misclassified as p-1-h LCO and two images as unknown. The proposed CNN model's performance metric for cross-domain experiment 3 is shown in Table13.

D. EXPERIMENT 4: CROSS-DOMAIN CLASSIFICATION OF DYNAMICAL STATES USING CNN MODEL TRAINED ON AEROELASTIC SYSTEM WITH CUBIC NONLINEARITY DATASET.

In this experiment, we train the CNN model with the data from aeroelastic system with cubic nonlinearity (source domain) and employ it to detect dynamical states pertinent to other datasets (Table 2). This experiment's source and target domains share two dynamical states, fixed point and LCO. On the other hand, chaos, period-doubling, p-1-h LCO, p-2-h LCO, and aperiodic LCO are not present in the source domain. These dynamical states, labelled as unknown states, were trained alongside subsets of dynamical states present in the source domain. Table 14 shows the familiar and unknown dynamical states in experiment 4. During training, a batch size of 10 was considered, and the model was trained for 200 epochs. The model achieves 98% training accuracy, and an accuracy of 100% is achieved in the validation result. The training accuracy and loss plots are shown in Figure 21 and 20. Between the train and test curves, it is seen

TABLE 14. Common and unknown dynamical states in cross-domain experiment 4.

Source domain	Target domain			Common dynamical states in source & target domain	Unknown dynamical states in source domain
Aeroelastic system with cubic nonlinearity	Pulse combustor	Aeroelastic system with freeplay nonlinearity	Aeroelastic system with combined freeplay and aerodynamic nonlinearity		
LCO	Chaos	Chaos	Aperiodic LCO	LCO	Chaos
Fixed point	Period-doubling	p-1-h LCO	LCO	Fixed point	Period-doubling
	LCO	p-2-h LCO	p-1-h LCO		p-1-h LCO
	Fixed point	LCO	Fixed point		p-2-h LCO
		Fixed point			Aperiodic LCO

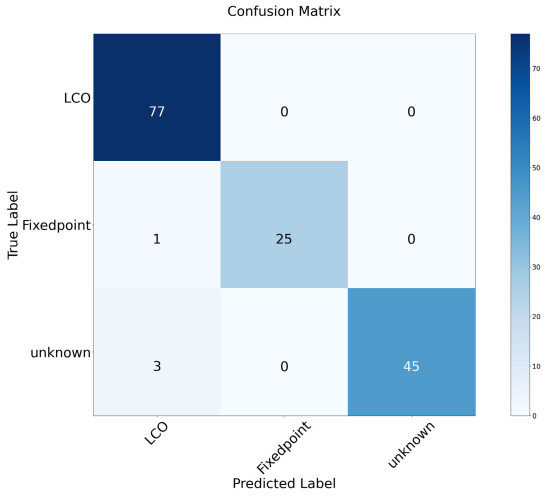

FIGURE 19. Confusion matrix for cross-domain experiment 4.

TABLE 15. Performance metric of the proposed CNN model for cross-domain experiment 4.

Dynamical States	Prescision	Recall	F1-Score
LCO	0.95	1.00	0.97
Fixed point	1.00	0.96	0.98
Unknown	1.00	0.94	0.97

that there is little difference. That way, we can be certain that our suggested model did not overfit. The confusion matrix in Figure 19 demonstrates that the proposed CNN model achieves an impressive test accuracy of 97%. In K-fold cross-validation, the model maintains an average test accuracy of 97.60 (+- 1.02) across all folds. Analyzing the confusion matrix, we note that the CNN excels with 100% accuracy in identifying fixed points. It also performs well, with 96% accuracy, in classifying LCO and achieving 94% accuracy in identifying unknown states. In this experiment, it is observed that the model misclassifies three images of an unknown dynamical state and one image of a fixed point as LCO. The proposed CNN model's performance metric for cross-domain experiment 4 is shown in Table 15.

E. INFLUENCE OF DEEP CNN MODELS ON CLASSIFICATION PERFORMANCE

In this experiment, we conducted a comparison of the classification performance between deep CNN models and the proposed CNN model. We carried out a series of

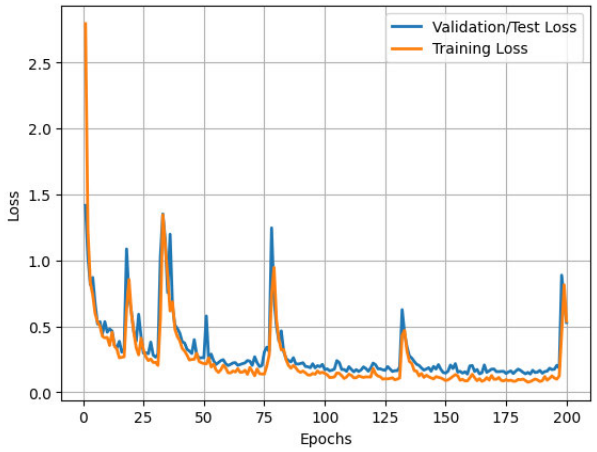

FIGURE 20. Training loss plot of experiment 4.

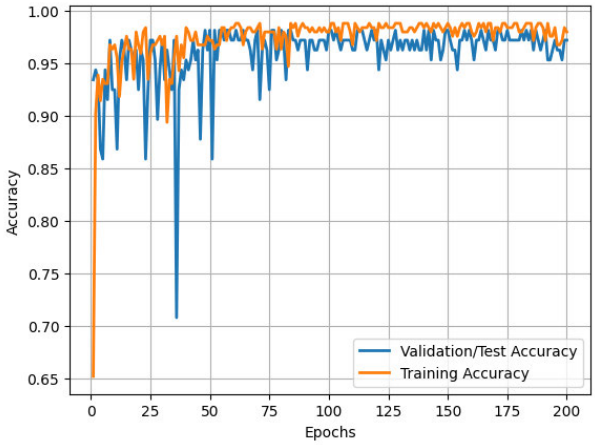

FIGURE 21. Training accuracy plot of cross-domain experiment 4.

comprehensive experiments, increasing the number of feature extraction layers from 3 to 5 for all cross-domain experiments. The details of the CNN models are illustrated in the Tables 16, 17, and 18. Importantly, the hyperparameters of the feature extraction layers for these CNN models were kept consistent with the ones used in the proposed CNN model introduced in this study. The classification performance of the deep CNN models was assessed using various metrics such as precision, recall, and F1-score. The Tables 19, 20, 21, and 22 illustrate the classification evaluation metrics for

TABLE 16. CNN2 model architecture.

Layers	Type	Subtype	Configuration	Activation
1	Input Layer	Image Input	256 x 256 x 1 images	256 x 256 x 1
2	1st Feature Extraction Layer	Convolution	32 convolutions: 3x3x1 filters, stride 1, 'same' padding	256 x 256 x 32
3		ReLU	ReLu	256 x 256 x 32
4		Max Pooling	2x2 max-pooling: stride 2, no padding	128 x 128 x 32
5	2nd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	128 x 128 x 32
6		Relu	ReLu	128 x 128 x 32
7		Max Pooling	2x2 max-pooling: stride 2, no padding	64 x 64 x 32
8	3rd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	64 x 64 x 32
9		Relu	ReLu	64 x 64 x 32
10		Max Pooling	2x2 max-pooling: stride 2, no padding	32 x 32 x 32
11	Classification Layer	Dropout	50% dropout	32 x 32 x 32
12		Fully Connected	1 dense layer with 128 neurons	128
13		Softmax	softmax	
14	Output Layer	Classification Output	No of classes	

TABLE 17. CNN3 model architecture.

Layers	Type	Subtype	Configuration	Activation
1	Input Layer	Image Input	256 x 256 x 1 images	256 x 256 x 1
2	1st Feature Extraction Layer	Convolution	32 convolutions: 3x3x1 filters, stride 1, 'same' padding	256 x 256 x 32
3		ReLU	ReLu	256 x 256 x 32
4		Max Pooling	2x2 max-pooling: stride 2, no padding	128 x 128 x 32
5	2nd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	128 x 128 x 32
6		Relu	ReLu	128 x 128 x 32
7		Max Pooling	2x2 max-pooling: stride 2, no padding	64 x 64 x 32
8	3rd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	64 x 64 x 32
9		Relu	ReLu	64 x 64 x 32
10		Max Pooling	2x2 max-pooling: stride 2, no padding	32 x 32 x 32
11	4th Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	32 x 32 x 32
12		Relu	ReLu	32 x 32 x 32
13		Max Pooling	2x2 max-pooling: stride 2, no padding	16 x 16 x 32
14	Classification Layer	Dropout	50% dropout	16 x 16 x 32
15		Fully Connected	1 dense layer with 128 neurons	128
16		Softmax	softmax	
17	Output Layer	Classification Output	No of classes	

TABLE 18. CNN4 model architecture.

Layers	Type	Subtype	Configuration	Activation
1	Input Layer	Image Input	256 x 256 x 1 images	256 x 256 x 1
2	1st Feature Extraction Layer	Convolution	32 convolutions: 3x3x1 filters, stride 1, 'same' padding	256 x 256 x 32
3		ReLU	ReLu	256 x 256 x 32
4		Max Pooling	2x2 max-pooling: stride 2, no padding	128 x 128 x 32
5	2nd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	128 x 128 x 32
6		Relu	ReLu	128 x 128 x 32
7		Max Pooling	2x2 max-pooling: stride 2, no padding	64 x 64 x 32
8	3rd Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	64 x 64 x 32
9		Relu	ReLu	64 x 64 x 32
10		Max Pooling	2x2 max-pooling: stride 2, no padding	32 x 32 x 32
11	4th Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	32 x 32 x 32
12		Relu	ReLu	32 x 32 x 32
13		Max Pooling	2x2 max-pooling: stride 2, no padding	16 x 16 x 32
14	5th Feature Extraction Layer	Convolution	32 convolutions: 3x3x32 filters, stride 1, 'same' padding	16 x 16 x 32
15		Relu	ReLu	16 x 16 x 32
16		Max Pooling	2x2 max-pooling: stride 2, no padding	8 x 8 x 32
17	Classification Layer	Dropout	50% dropout	8 x 8 x 32
18		Fully Connected	1 dense layer with 128 neurons	128
19		Softmax	softmax	
20	Output Layer	Classification Output	No of classes	

cross-domain experiments 1 to 4. Figure 22 presents the classification performance of the proposed CNN model alongside deeper CNN models. It is evident from the figure that the investigation revealed that deeper CNN models did not significantly improve the classification accuracy for our dataset. Our results demonstrate that our proposed shallower model achieved competitive and satisfactory performance.

TABLE 19. Comparison of classification evaluation metrics for proposed CNN and other CNN models for cross-domain experiment 1.

	CNN (proposed model)			CNN 2			CNN 3			CNN 4		
Dynamical States	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	Precision
Chaos	1.00	1.00	1.00	0.96	0.96	0.96	1.00	1.00	1.00	1.00	0.96	0.98
Period-doubling	1.00	1.00	1.00	0.88	1.00	0.94	1.00	1.00	1.00	0.92	0.80	0.86
LCO	1.00	0.97	0.99	0.99	0.96	0.97	0.99	0.97	0.98	0.91	0.99	0.95
Fixedpoint	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Unknown	0.86	1.00	0.92	0.87	0.87	0.87	0.88	0.93	0.90	1.00	0.73	0.85

TABLE 20. Comparison of classification evaluation metrics for proposed CNN and other CNN models for cross-domain experiment 2.

	CNN (proposed model)			CNN 2			CNN 3			CNN 4		
Dynamical States	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	Precision
Chaos	1.00	1.00	1.00	0.89	1.00	0.94	0.94	0.94	0.94	0.94	1.00	0.97
p-1-h LCO	0.91	0.91	0.91	0.45	0.83	0.59	0.45	0.83	0.59	0.44	0.67	0.53
LCO	0.99	1.00	0.99	1.00	0.94	0.97	1.00	0.94	0.97	0.99	0.95	0.97
Fixedpoint	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
p-2-h LCO	0.25	0.50	0.33	0.67	0.67	0.67	0.67	0.67	0.67	1.00	0.67	0.80
Unknown	0.86	1.00	0.92	0.87	0.87	0.87	0.88	0.93	0.90	1.00	0.73	0.85

TABLE 21. Comparison of classification evaluation metrics for proposed CNN and other CNN models for cross-domain experiment 3.

	CNN (proposed model)			CNN 2			CNN 3			CNN 4		
Dynamical States	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	Precision
Aperiodic LCO	1.00	1.00	1.00	0.29	1.00	0.44	0.75	0.60	0.67	0.00	0.00	0.00
p-1-h LCO	0.86	0.67	0.75	0.67	0.86	0.75	0.50	0.83	0.62	0.56	0.99	0.72
LCO	0.96	0.96	0.96	0.99	0.96	0.98	0.99	0.92	0.95	0.96	1.00	0.98
Fixedpoint	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00
Unknown	0.95	1.00	0.98	0.97	0.85	0.91	0.91	0.96	0.93	0.00	0.00	0.00

TABLE 22. Comparison of classification evaluation metrics for proposed CNN and other CNN models for cross-domain experiment 4.

	CNN (proposed model)			CNN 2			CNN 3			CNN 4		
Dynamical States	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	Precision
LCO	0.95	1.00	0.97	0.95	1.00	0.97	0.96	1.00	0.98	0.96	1.00	0.98
Fixedpoint	1.00	0.96	0.98	1.00	0.96	0.98	1.00	1.00	1.00	1.00	1.00	1.00
Unknown	1.00	0.94	0.97	1.00	0.94	0.97	1.00	0.94	0.97	1.00	0.94	0.97

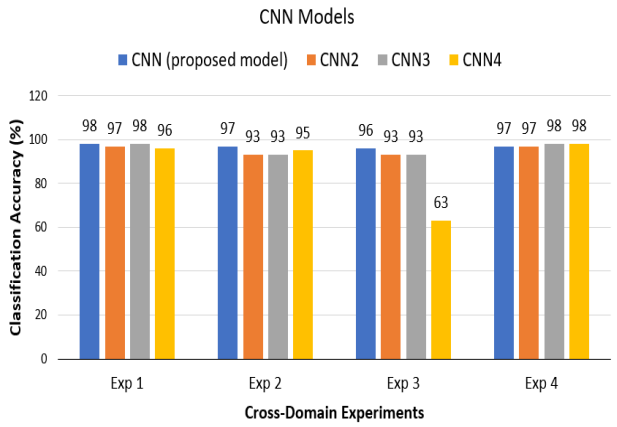

FIGURE 22. Classification accuracies of the CNN models across four cross-domain Experiments.

F. COMPARING STFT AND RPs FOR CNN CLASSIFICATION

This section compares classification accuracies between our proposed CNN model, utilizing RPs as input, and another common pre-processing technique, Short Time Fourier Transform (STFT). STFT is commonly utilized in various fields, including signal processing [80], audio analysis [81] and other machine learning applications, to transform time series data into visual representations. This experiment aims to assess whether STFT representation of the times series pertaining to the dynamical states of the nonlinear fluid dynamical systems can improve or show similar classification performance to RPs. This section presents the classification results of our proposed CNN model with spectrogram as inputs for classifying different dynamical states of non-linear fluid dynamical systems for all four cross-domain experiments.

1) GENERATING SPECTROGRAM IMAGES:

A spectrogram visualizes a time series against frequencies, showcasing the signal's strength at specific time instances. It is presented as a two-dimensional graph, with frequency plotted along the vertical axis and time along the horizontal axis. In this study, STFT images are generated for each time series corresponding to a dynamical state (after initial transients are removed) in the dataset. This is achieved using the 'specgram' function from the Matplotlib module in Python. The 'specgram' function uses 'viridis' as a default colormap. The intensity of the magnitudes of the frequencies is represented using shades of yellow to green, where dark regions (green) indicate less intense frequency

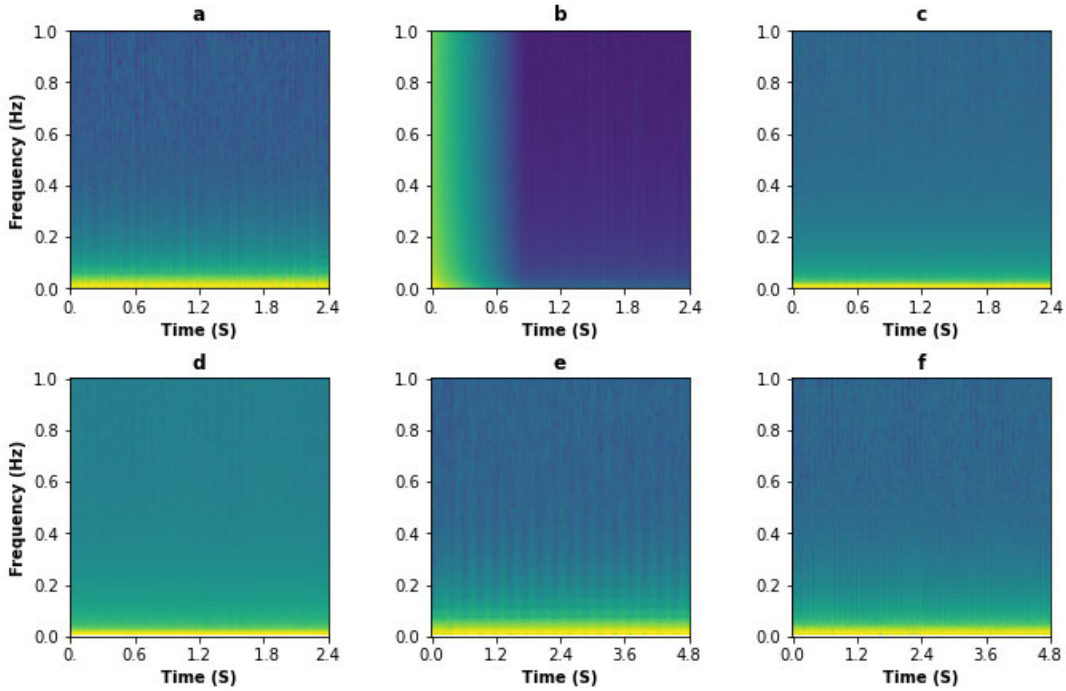

FIGURE 23. Sample spectrogram images of dynamical states: (a) Chaos, (b) Fixed point, (c) LCO, (d) Period-doubling, (e) p-1-h LCO, (f) p-2-h LCO.

components, and bright regions (yellow) indicate more intense frequency components. Subsequently, these images are resized to dimensions of 256 × 256 pixels, standardized, and utilized as input for the CNN classifier. A sample of these Short-Time Fourier Transform (STFT) images depicting various dynamical states of the considered nonlinear fluid dynamical systems is presented in Figure 23. As observed in Figure (a) ('Chaos'), the spectrogram captures the complex and irregular behavior associated with chaotic dynamics. The broad frequency distribution and lack of clear patterns signify the inherent unpredictability in chaotic time series. In Figure (b) ('Fixed Point'), the spectrogram exhibits consistent frequency and intensity up to 0.6 seconds. After 0.6 seconds, the intensity of the frequency diminishes, indicating a transition to a steady-state dynamic. In Figures (c) ('LCO'), (d) ('Period-doubling), and (f) ('p-2-h lco'), it is observed that the spectrograms captures repeating frequency components, emphasizing the presence of recurrent frequency patterns. In Figure (e) (p-1-h LCo), the spectrogram highlights the presence of a period 1-harmonic limit cycle oscillation with a layered frequency pattern. The model's hyperparameters and architecture remain consistent with those employed for the RPs approach (Table 7). Following this, the OS domain adaptation experiment is executed for four cross-domain experiments (Table 2).

2) SPECTROGRAM-BASED CLASSIFICATION PERFORMANCE:

Tables 23 to 26 illustrates the evaluation metrics for cross-domain experiments 1 to 4. Figures 25 to 28 shows the confusion matrices for cross-domain experiments 1 to 4.

As seen from the Figure 26, for cross-domain experiment 1 (Table 8) the model performs exceptionally well in identifying Fixed Point, achieving a 100% accuracy rate. Additionally, the model demonstrates solid accuracy in recognizing LCO with an 87% accuracy, and Period-doubling with an 88% accuracy. However, the model encounters some challenges when it comes to identifying Unknown dynamical states, achieving a 67% accuracy. Similarly, the accuracy drops to 48% when attempting to identify Chaos states.

Referring to Figure 27, for cross-domain experiment 2 (Table 10) we see that the model excels in classifying the Fixed Point dynamical state with an impressive 95% accuracy. It also demonstrates strong accuracy in recognizing LCO at 87%, and p-1-h LCO at 82%. When it comes to p-2-h LCO, the model maintains a commendable 75% accuracy. However, the model faces some challenges when dealing with Unknown dynamical states, achieving a modest 20% accuracy. Similarly, its accuracy drops to 34% when attempting to identify Chaos states.

As can be seen from the Figure 28, for cross-domain experiment 3 (Table 12) we can observe that the model achieves a perfect 100% accuracy in classifying the Fixed Point. Additionally, the model performs well in identifying Unknown dynamical states, with an 88% accuracy. However, the model faces some difficulties when it comes to LCO, achieving a 68% accuracy. Similarly, it achieves a 67% accuracy for Aperiodic LCO. The model's accuracy drops further, reaching 34%, when attempting to identify p-1-h LCO.

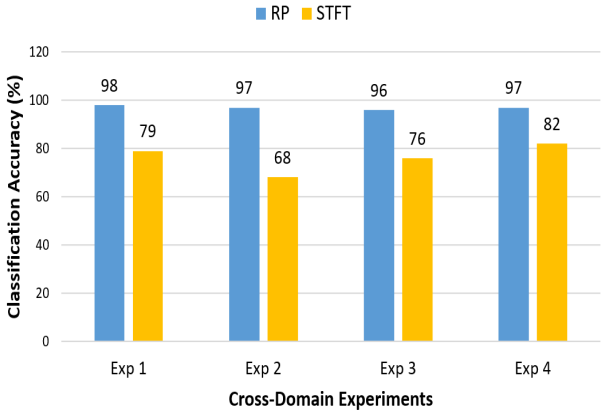

FIGURE 24. comparing classification accuracies of proposed CNN model with RP and STFT as inputs across four cross-domain experiments.

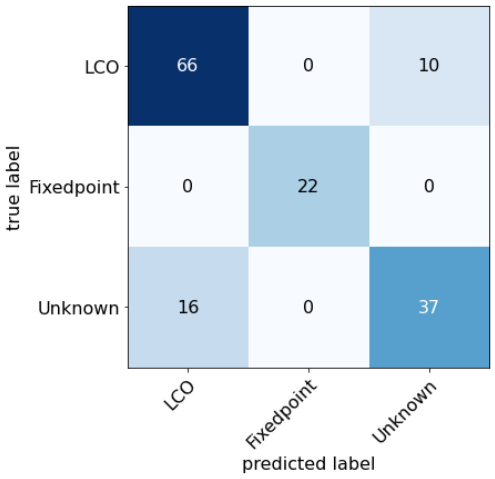

FIGURE 25. Confusion matrix of CNN with STFT as input for cross-domain experiment 4.

Referring to Figure 25, for cross-domain experiment 4 (Table 14) the model achieves a remarkable 100% accuracy in classifying the Fixed Point. Furthermore, the model impressively identifies LCO with an 87% accuracy rate. However, its accuracy decreases to 70% when trying to identify Unknown dynamical state.

Figure 24 presents a comparison of classification accuracies between our proposed CNN model with recurrence plot as input and with spectrogram as input for all four cross-domain experiments. As can be seen for the Figure 24, the model utilizing recurrence plots achieves the highest classification accuracy in all experiments. Furthermore, the model's accuracy consistently outperforms when trained with RPs as inputs compared to STFT inputs. This observation strongly emphasizes the efficacy of recurrence plots in capturing essential patterns and information, contributing to accurate classification within the specific context of dynamical state classification.

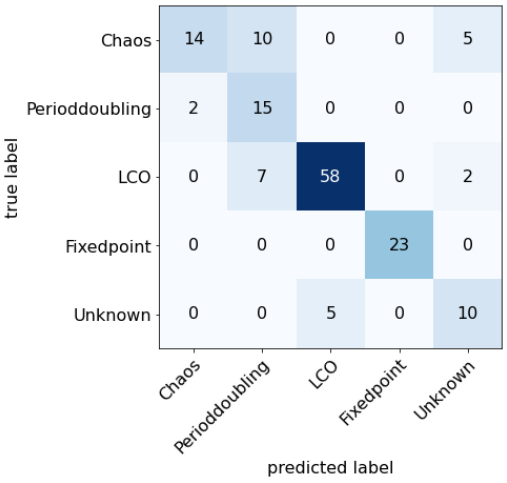

FIGURE 26. Confusion matrix of CNN with STFT as input for cross-domain experiment 1.

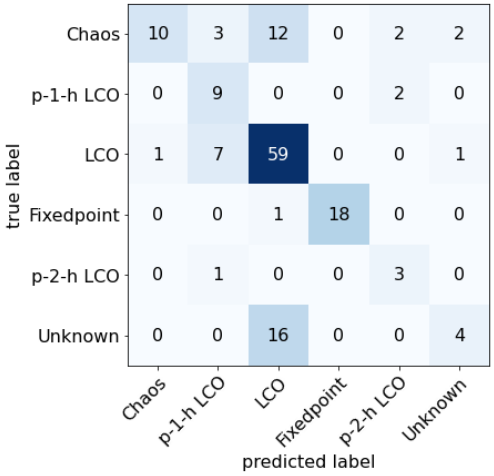

FIGURE 27. Confusion matrix of CNN with STFT as input for cross-domain experiment 2.

TABLE 23. Performance metric of CNN using STFT for cross-domain experiment 1.

Dynamical States	Precision	Recall	F1-Score
Chaos	0.88	0.48	0.62
Perioddoubling	0.47	0.88	0.61
LCO	0.92	0.87	0.89
Fixedpoint	1.00	1.00	1.00
Unknown	0.59	0.67	0.62

G. VISUALIZING CNN MODEL INTERPRETABILITY WITH GRAD-CAM ANALYSIS

In this section, in order to understand the outcomes of the suggested model, we used the Grad-CAM technique to visually represent the output of the final convolution layer of the CNN model. The technique is proposed by Selvaraju et al. [82] to highlight significant regions of the image that are class-specific, making the black box

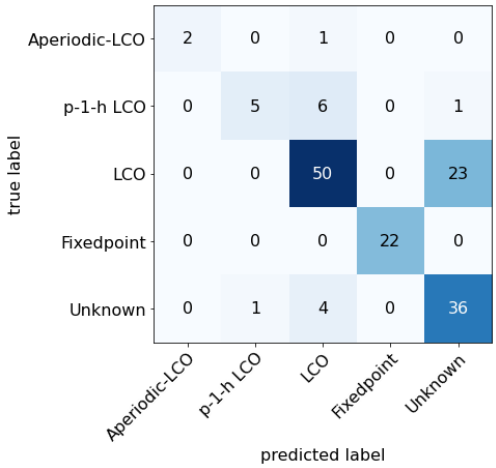

FIGURE 28. Confusion matrix of CNN with STFT as input for cross-domain experiment 3.

TABLE 24. Performance metric of CNN using STFT for cross-domain experiment 2.

Dynamical States	Precision	Recall	F1-Score
Chaos	0.91	0.34	0.50
p-1-h LCO	0.45	0.82	0.58
LCO	0.67	0.87	0.76
Fixedpoint	1.00	0.95	0.97
p-2-h LCO	0.43	0.75	0.55
Unknown	0.57	0.20	0.30

TABLE 25. Performance metric of CNN using STFT for cross-domain experiment 3.

Dynamical States	Precision	Recall	F1-Score
Aperiodic-LCO	1.00	0.67	0.80
p-1-h LCO	0.83	0.42	0.56
LCO	0.82	0.68	0.75
Fixedpoint	1.00	1.00	1.00
Unknown	0.60	0.88	0.71

TABLE 26. Performance metric of CNN using STFT for cross-domain experiment 4.

Dynamical States	Precision	Recall	F1-Score
LCO	0.80	0.87	0.84
Fixedpoint	1.00	1.00	1.00
Unknown	0.79	0.70	0.74

TABLE 27. SVM hyperparameters.

Hyperparameters	Range of Values			
C (penalty parameter)	0.1	1	10	100
Gamma (kernel coefficient)	0.0001	0.001	0.1	1
Kernel	rbf		poly	

models (Deep Learning Models) more transparent. Grad-CAM extracts the last convolution layer's feature maps and is subjected to a weighted sum to yield a Grad-CAM heat map. Then heat map is overlayed with the input image with shades

TABLE 28. SVM classification metrics for cross-domain experiment 2.

Dynamical States	Precision	Recall	F1-Score
Chaos	0.82	0.93	0.87
p-1-h LCO	0.00	0.00	0.00
LCO	0.96	1.00	0.98
Fixedpoint	1.00	1.00	1.00
p-2-h LCO	0.00	0.00	0.00
Unknown	0.86	1.00	0.92

TABLE 29. SVM classification metrics for cross-domain experiment 1.

Dynamical States	Precision	Recall	F1-Score
Chaos	0.83	1.00	0.91
Perioddoubling	0.85	1.00	0.92
LCO	0.98	0.96	0.97
Fixedpoint	1.00	1.00	1.00
Unknown	0.60	0.33	0.43

TABLE 30. SVM classification metrics for cross-domain experiment 3.

Dynamical States	Precision	Recall	F1-Score
Aperiodic-LCO	0.00	0.00	0.00
p-1-h LCO	0.67	0.40	0.50
LCO	0.96	0.96	0.96
Fixedpoint	1.00	1.00	1.00
Unknown	0.90	1.00	0.95

TABLE 31. SVM classification metrics for cross-domain experiment 4.

Dynamical States	Precision	Recall	F1-Score
LCO	1.00	0.96	0.98
Fixedpoint	1.00	1.00	1.00
Unknown	0.95	1.00	0.97

of red to green. The red region denotes the most emphasized region, and green region denotes the least emphasized region to classify an image. A sample of the results of the Grad-CAM analysis is illustrated in Figure 29. It is shown that the patterns corresponding to the recurrence plots for all classes namely chaos, period-doubling, LCO, fixed point, aperiodic LCO, p-1-h LCO, and p-2-h LCO are captured by the proposed CNN architecture.

H. COMPARING CNN AND SVM PERFORMANCE IN DYNAMICAL STATE CLASSIFICATION

In this section, we compared the classification performance of the proposed CNN model for classifying the dynamical states of nonlinear fluid dynamical systems with SVM. This comprehensive experiment was assessed across all four cross-domain experiments presented in our study.

To implement the SVM model, we employed the GridSearchCV algorithm. This technique allowed us to fine-tune the model's hyperparameters, including C (penalty parameter), gamma (kernel coefficient), and the type of kernel used. By setting ranges for individual hyperparameters, the GridSearchCV algorithm systematically explored various hyperparameter combinations to find the most suitable configuration. The hyperparameter ranges selected for this experiment were presented in Table 27. The data was

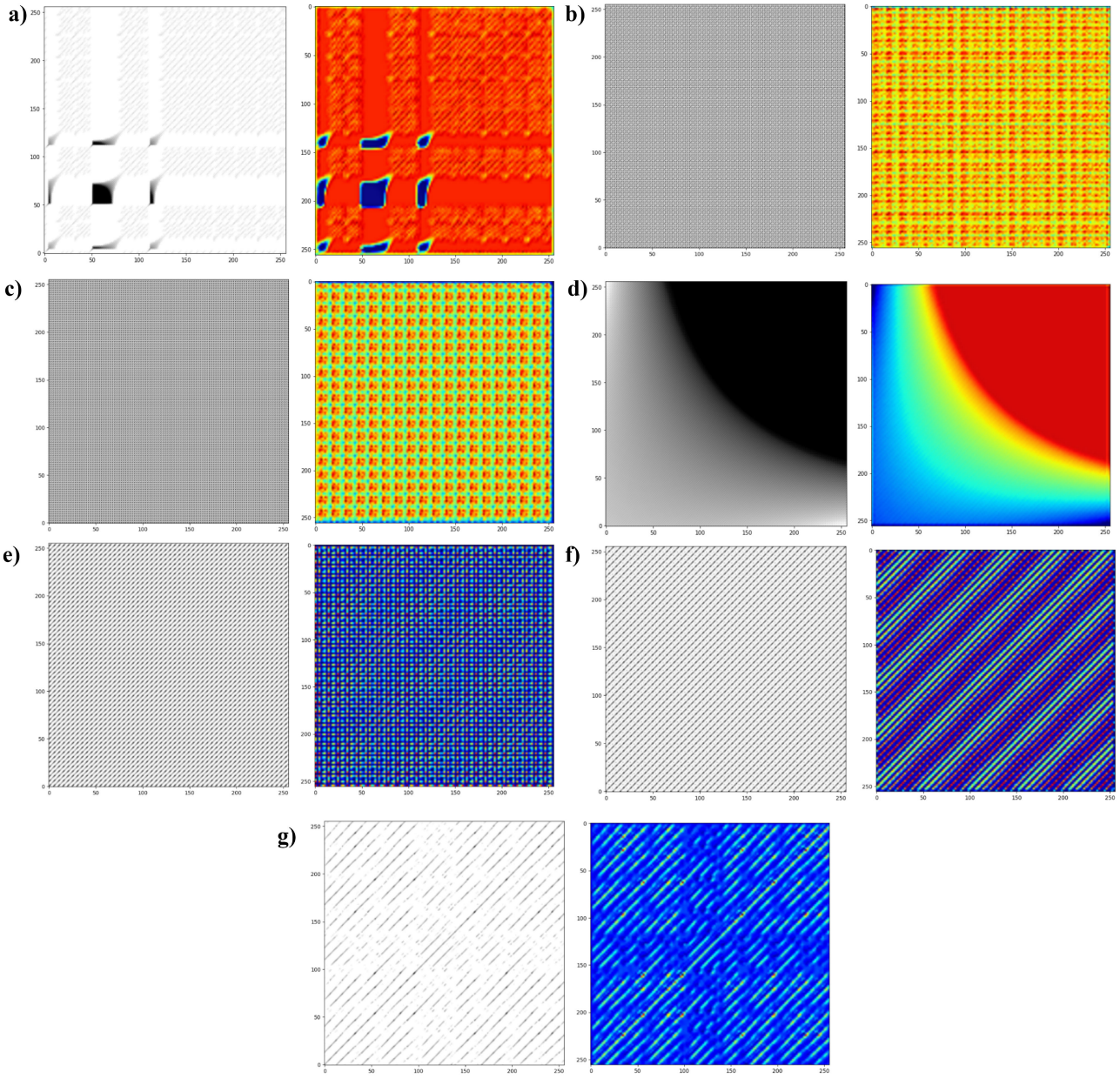

FIGURE 29. A sample of Grad-CAM analysis on different dynamical states. a) Chaos, b) Period-doubling, c) LCO, d) Fixed point, e) p-1-h LCO, f) p-2-h LCO, and g) Aperiodic LCO.

split into training and testing data in an 80/20 split. The classification metrics were evaluated based on precision, recall, and F1 score. Tables 29, 28, 30, and 31 illustrates the evaluation metrics for cross-domain experiments 1 to 4. Figures 30, 31, 32, and 33 shows the confusion matrices for cross-domain experiments 1 to 4.

As seen from the Figure 30, for cross-domain experiment 1 (Table 8) SVM model achieves 100% accuracy in classifying dynamical states Chaos, Period doubling, and Fixed point. The model also shows 96% accuracy in identifying LCO. However, the model only achieves 33% accuracy in identifying unknown dynamical states.

As seen from the Figure 31, for cross-domain experiment 2 (Table 10) SVM model achieves 100% accuracy in classifying dynamical states LCO, Unknown, and Fixed point. The model also shows 93% accuracy in identifying Chaos. However, the model completely misclassifies the dynamical states p-1-h LCO and p-2-h LCO.

As observed in Figure 32, for cross-domain experiment 3 (Table 12) the SVM model achieves 100% accuracy in classifying the dynamical states Unknown and Fixed Point. It achieves an accuracy of 96% in identifying LCO. Additionally, the model attains a 40% accuracy in recognizing p-1-h LCO. However, the model encounters challenges in

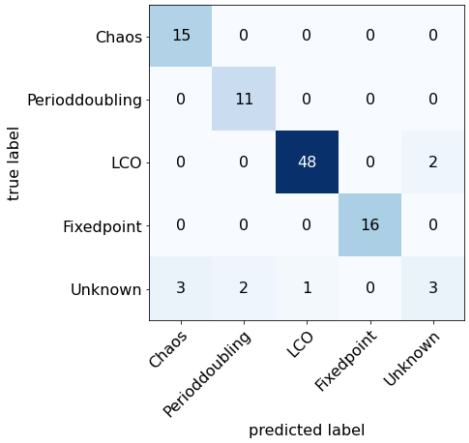

FIGURE 30. SVM confusion matrix for cross-domain experiment 1.

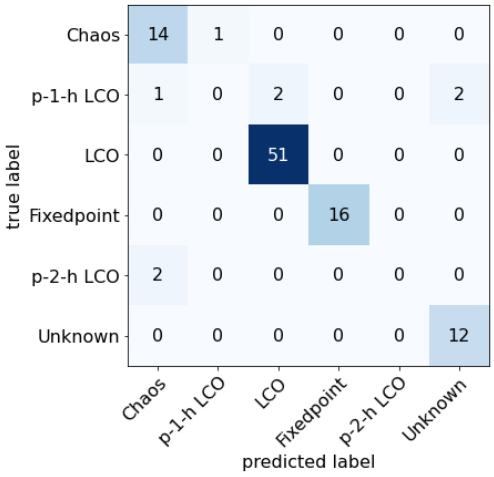

FIGURE 31. SVM confusion matrix for cross-domain experiment 2.

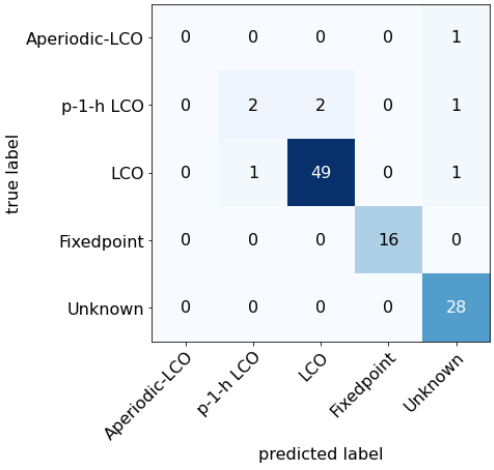

FIGURE 32. SVM confusion matrix for cross-domain experiment 3.

accurately identifying the Aperiodic-LCO dynamical state, resulting in misclassification.

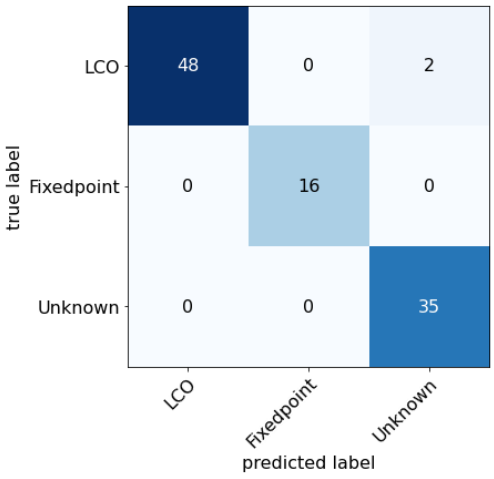

FIGURE 33. SVM confusion matrix for cross-domain experiment 4.

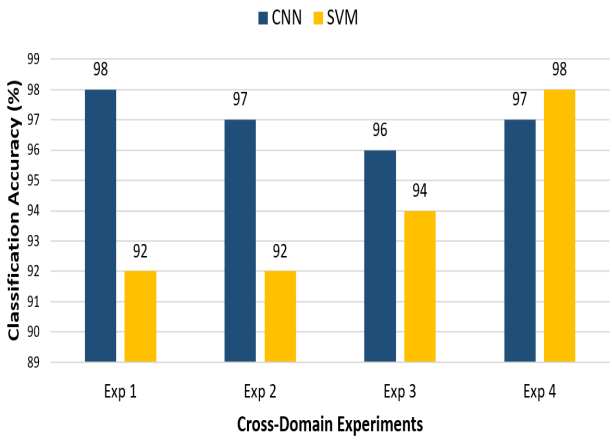

FIGURE 34. comparing classification accuracies of proposed CNN model with SVM.

As depicted in Figure 33, for cross-domain experiment 4 (Table 14) the SVM model achieves 100% accuracy in classifying the dynamical states Unknown and Fixed Point. Additionally, the model demonstrates a 96% accuracy rate in identifying the LCO.

Figure 34 shows the comparison of the classification accuracies of our proposed CNN model with SVM model for all four cross-domain experiments. As can be seen from Figure 34 CNN achieves highest classification accuracy in cross-domain experiments 1 to 3, on the other hand SVM achieves highest classification accuracy in cross-domain experiment 4. Though the SVM model shows excellent classification performance in classifying the dynamical states, it falls short of the performance of our suggested CNN model.

V. CONCLUDING REMARKS

This paper introduces an open-set (OS) domain adaptation-based deep learning technique for classifying dynamical states of nonlinear fluid dynamical systems.Nonlinear fluid

dynamical systems are susceptible to unexpected and sometimes catastrophic bifurcations as a result of gradual changes in system parameters. These changes can trigger state transitions that unfold in specific directions, invariably regarded as undesirable, and hold the capacity to profoundly modify the operational characteristics of these systems. Therefore, distinguishing the qualitative nature of the dynamical state within the nonlinear fluid dynamical system is paramount in preventing catastrophic state transitions. In this study, we present an automated approach for the classification of dynamical states of nonlinear fluid dynamical systems using deep learning. Furthermore, given that numerous nonlinear fluid dynamical systems can exhibit similar bifurcation scenarios, we introduce the concept of open-set domain adaptation (OS). With OS, a model initially trained to classify dynamical states in one specific nonlinear fluid dynamical system can be extended to classify dynamical states in other nonlinear fluid dynamical systems by making appropriate adjustments to the deep learning model. This innovative approach eliminates the need to train a new model for each individual nonlinear fluid dynamical system. To the best of our knowledge, this study is the first to use an OS adaptation-based deep learning technique for the classification of dynamical states in nonlinear fluid dynamical systems.

We have proposed the use of image representations derived from time series data associated with the dynamical states of nonlinear fluid dynamical systems, reframing the problem as a pattern recognition task employing deep learning. Instead of directly using raw time series data, our approach involves acquiring knowledge about the dynamical system through image-based features. We have developed a convolutional neural network (CNN) for the multi-class classification of these dynamical states within nonlinear fluid dynamical systems, drawing inspiration from the successful application of deep learning in automated feature extraction and image classification. Our proposal involves leveraging the dynamic data through textured recurrence plot (RP) images, which depict the time-series characteristics as textured 2D images. These images serve as a visual representation of the time series, aiding in the differentiation of dynamical states.

Experiments were conducted on four source-target domain combinations to assess the model's ability to generalize to unfamiliar data while maintaining a consistent model architecture. The experimental results consistently demonstrated the model's impressive performance, achieving above 95% accuracy across all conducted experiments. Further investigations were carried out to establish the efficacy of Recurrence Plots (RPs) in capturing essential patterns and information crucial for accurate classification, in comparison to the widely used pre-processing technique Short-Time Fourier Transform (STFT). These experiments conclusively revealed that the CNN model utilizing RPs as input consistently outperformed the CNN model employing STFT representations of the time series data corresponding to the dynamical states. These findings underline the superiority of RPs in enhancing the classification process.

Moreover, the classification performance of the proposed CNN model employing RPs as input was compared with SVM across all cross-domain experiments. Experimental results show that the proposed CNN model consistently outperforms SVM. Although the results are promising, this study primarily addresses the OS domain adaptation problem, where the source and target domains have only a few shared dynamical states. Additionally, the dynamical states in both the source and target domains are manually labeled, which is a laborious and time-consuming process. In real-world scenarios requiring model deployment, labeled data may be exclusive to the source domain, or both source and target domains could contain unlabeled data. The application of the proposed method to such scenarios poses challenges. Future research will shift its focus to unsupervised dynamical state classification, which is more practical for engineering applications.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, H. Held, E. H. van Nes, M. Rietkerk, and G. Sugihara, "Early-warning signals for critical transitions," *Nature*, vol. 461, no. 7260, pp. 53–59, Sep. 2009.

[2] D. Hodgson, J. L. McDonald, and D. J. Hosken, "What do you mean, 'resilient'?" *Trends Ecol. Evol.*, vol. 30, no. 9, pp. 503–506, 2015.

[3] D. Tripathi, R. Shreenivas, C. Bose, S. Mondal, and J. Venkatramani, "Experimental investigation on the synchronization characteristics of a pitch-plunge aeroelastic system exhibiting stall flutter," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 32, no. 7, Jul. 2022, Art. no. 073114.

[4] G. Dimitriadis and J. Li, "Bifurcation behavior of airfoil undergoing stall flutter oscillations in low-speed wind tunnel," *AIAA J.*, vol. 47, no. 11, pp. 2577–2596, Nov. 2009.

[5] J. Venkatramani, S. Sarkar, and S. Gupta, "Intermittency in pitch-plunge aeroelastic systems explained through stochastic bifurcations," *Nonlinear Dyn.*, vol. 92, no. 3, pp. 1225–1241, May 2018.

[6] G. Dimitriadis, *Introduction to Nonlinear Aeroelasticity*. Hoboken, NJ, USA: Wiley, 2017.

[7] W. V. Wedig, "Dynamic stability of beams under axial forces–Lyapunov exponents for general fluctuating loads," in *Structural Dynamics*. Evanston, IL, USA: Routledge, 2022, pp. 141–148.

[8] P. Kumar, S. Narayanan, and S. Gupta, "Stochastic bifurcations in a vibro-impact Duffing–Van der Pol oscillator," *Nonlinear Dyn.*, vol. 85, no. 1, pp. 439–452, Jul. 2016.

[9] C. C. Marsden and S. J. Price, "The aeroelastic response of a wing section with a structural freeplay nonlinearity: An experimental investigation," *J. Fluids Struct.*, vol. 21, no. 3, pp. 257–276, Nov. 2005.

[10] S. Mondal, A. Mukhopadhyay, and S. Sen, "Bifurcation analysis of steady states and limit cycles in a thermal pulse combustor model," *Combustion Theory Model.*, vol. 21, no. 3, pp. 487–502, May 2017.

[11] S. Datta, S. Mondal, A. Mukhopadhyay, D. Sanyal, and S. Sen, "An investigation of nonlinear dynamics of a thermal pulse combustor," *Combustion Theory Model.*, vol. 13, no. 1, pp. 17–38, Jan. 2009.
[12] J. Venkatramani, V. Nair, R. I. Sujith, S. Gupta, and S. Sarkar, "Multi-fractality in aeroelastic response as a precursor to flutter," *J. Sound Vib.*, vol. 386, pp. 390–406, Jan. 2017.
[13] V. Nair, G. Thampi, and R. I. Sujith, "Intermittency route to thermoacoustic instability in turbulent combustors," *J. Fluid Mech.*, vol. 756, pp. 470–487, Oct. 2014.
[14] J. Venkatramani, V. Nair, R. I. Sujith, S. Gupta, and S. Sarkar, "Precursors to flutter instability by an intermittency route: A model free approach," *J. Fluids Struct.*, vol. 61, pp. 376–391, Feb. 2016.
[15] R. I. Sujith and V. R. Unni, "Complex system approach to investigate and mitigate thermoacoustic instability in turbulent combustors," *Phys. Fluids*, vol. 32, no. 6, Jun. 2020, Art. no. 061401.
[16] X. Meng, W. de Jong, and T. Kudra, "A state-of-the-art review of pulse combustion: Principles, modeling, applications and R&D issues," *Renew. Sustain. Energy Rev.*, vol. 55, pp. 73–114, Mar. 2016.
[17] B. H. K. Lee, S. J. Price, and Y. S. Wong, "Nonlinear aeroelastic analysis of airfoils: Bifurcation and chaos," *Prog. Aerosp. Sci.*, vol. 35, no. 3, pp. 205–334, Apr. 1999.
[18] L. Kabiraj, A. Saurabh, P. Wahi, and R. I. Sujith, "Route to chaos for combustion instability in ducted laminar premixed flames," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 22, no. 2, Jun. 2012, Art. no. 023129.
[19] L. Kabiraj and R. I. Sujith, "Nonlinear self-excited thermoacoustic oscillations: Intermittency and flame blowout," *J. Fluid Mech.*, vol. 713, pp. 376–397, Dec. 2012.
[20] S. Mondal, S. A. Pawar, and R. I. Sujith, "Synchronous behaviour of two interacting oscillatory systems undergoing quasiperiodic route to chaos," *Chaos: Interdiscipl. J. Nonlinear Sci.*, vol. 27, no. 10, Oct. 2017, Art. no. 103119.
[21] S. Mondal, A. Mukhopadhyay, and S. Sen, "Dynamic characterization of a laboratory-scale pulse combustor," *Combustion Sci. Technol.*, vol. 186, no. 2, pp. 139–152, Feb. 2014.
[22] S. Mondal, A. Mukhopadhyay, and S. Sen, "Effects of inlet conditions on dynamics of a thermal pulse combustor," *Combustion Theory Model.*, vol. 16, no. 1, pp. 59–74, Feb. 2012.
[23] S. Vishal, A. Raaj, and C. Bose, "Routes to synchronization in a pitch–plunge aeroelastic system with coupled structural and aerodynamic nonlinearities," *Int. J. Non-Linear Mech.*, vol. 135, Oct. 2021, Art. no. 103766.
[24] S. Rangarajan, D. Tripathi, and J. Venkatramani, "Non-normality and transient growth in stall flutter instability," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 33, no. 3, Mar. 2023, Art. no. 031103.
[25] N. A. Razak, T. Andrianne, and G. Dimitriadis, "Flutter and stall flutter of a rectangular wing in a wind tunnel," *AIAA J.*, vol. 49, no. 10, pp. 2258–2271, Oct. 2011.
[26] S. Vishal, A. Raaj, C. Bose, J. Venkatramani, and G. Dimitriadis, "Numerical investigation into discontinuity-induced bifurcations in an aeroelastic system with coupled non-smooth nonlinearities," *Nonlinear Dyn.*, vol. 108, no. 4, pp. 3025–3051, Jun. 2022.
[27] J. Schijve, "Fatigue damage in aircraft structures, not wanted, but tolerated?" *Int. J. Fatigue*, vol. 31, no. 6, pp. 998–1011, Jun. 2009.
[28] S. Venkatesh, S. Sarkar, and I. Rychlik, "Uncertainties in blade flutter damage prediction under random gust," *Probabilistic Eng. Mech.*, vol. 36, pp. 45–55, Apr. 2014.
[29] D. Tripathi, S. Vishal, C. Bose, and J. Venkatramani, "Stall-induced fatigue damage in nonlinear aeroelastic systems under stochastic inflow: Numerical and experimental analyses," *Int. J. Non-Linear Mech.*, vol. 142, Jun. 2022, Art. no. 104003.
[30] V. Nair and R. I. Sujith, "Multifractality in combustion noise: Predicting an impending combustion instability," *J. Fluid Mech.*, vol. 747, pp. 635–655, May 2014.
[31] S. Mondal, A. Mukhopadhyay, and S. Sen, "Non-linear dynamics in pulse combustor: A review," *Pramana*, vol. 84, no. 3, pp. 443–453, Mar. 2015.
[32] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.
[36] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero, "Recent advances in deep learning for speech research at Microsoft," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8604–8608.
[37] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[38] E. Haselsteiner and G. Pfurtscheller, "Using time-dependent neural networks for EEG classification," *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 457–463, Mar. 2000.
[39] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. neural Inf. Process. Syst.*, 2009, pp. 1096–1104.
[40] W. Yu, I. Y. Kim, and C. Mechefske, "Analysis of different RNN autoencoder variants for time series classification and machine prognostics," *Mech. Syst. Signal Process.*, vol. 149, Feb. 2021, Art. no. 107322.
[41] D. Wenkack Liedji, J. H. Talla Mbé, and G. Kenne, "Classification of hyperchaotic, chaotic, and regular signals using single nonlinear node delay-based reservoir computers," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 32, no. 12, Dec. 2022, Art. no. 123126.
[42] Y. Akkem, S. K. Biswas, and A. Varanasi, "Smart farming using artificial intelligence: A review," *Eng. Appl. Artif. Intell.*, vol. 120, Apr. 2023, Art. no. 105899.
[43] Y. Akkem, S. K. Biswas, and A. Varanasi, "Smart farming monitoring using ML and MLOPS," in *Proc. Int. Conf. Innov. Comput. Commun.* Cham, Switzerland: Springer, 2023, pp. 665–675.
[44] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10. Cambridge, MA, USA: MIT Press, 1995, p. 1995.
[45] N. Boullé, V. Dallas, Y. Nakatsukasa, and D. Samaddar, "Classification of chaotic time series with deep learning," *Phys. D, Nonlinear Phenomena*, vol. 403, Feb. 2020, Art. no. 132261.
[46] Z. Chen, K. Gryllias, and W. Li, "Mechanical fault diagnosis using convolutional neural networks and extreme learning machine," *Mech. Syst. Signal Process.*, vol. 133, Nov. 2019, Art. no. 106272.
[47] R. E. D. B. A. Barros and N. F. F. Ebecken, "Development of a ship classification method based on convolutional neural network and cyclostationarity analysis," *Mech. Syst. Signal Process.*, vol. 170, May 2022, Art. no. 108778.
[48] A. Ben-Cohen, R. Mechrez, N. Yedidia, and H. Greenspan, "Improving CNN training using disentanglement for liver lesion classification in CT," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 886–889.
[49] S. Srinivasan, V. Ravi, M. Krichen, D. B. Noureddine, and S. Anivilla, "Deep convolutional neural network based image spam classification," in *Proc. 6th Conf. Data Sci. Mach. Learn. Appl. (CDMA)*, Mar. 2020, pp. 112–117.
[50] T. K. S. Saj, S. Babu, V. K. Reddy, P. Gopika, V. Sowmya, and K. P. Soman, "Facial emotion recognition using shallow CNN," in *Machine Learning and Metaheuristics Algorithms, and Applications*. Cham, Switzerland: Springer, 2020, pp. 144–150.
[51] A. Kumar T. K., R. Vinayakumar, S. Variyar V. V., V. Sowmya, and K. P. Soman, "Convolutional neural networks for fingerprint liveness detection system," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICCS)*, May 2019, pp. 243–246.
[52] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhys. Lett.*, vol. 4, no. 9, pp. 973–977, Nov. 1987.
[53] S. Mukhopadhyay and S. Banerjee, "Learning dynamical systems in noise using convolutional neural networks," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 10, Oct. 2020, Art. no. 103125.
[54] G. Lee, D. Kwon, and C. Lee, "A convolutional neural network model for SOH estimation of Li-ion batteries with physical interpretability," *Mech. Syst. Signal Process.*, vol. 188, Apr. 2023, Art. no. 110004.

[55] Z. Zhao, Y. Zhang, Z. Comert, and Y. Deng, "Computer-aided diagnosis system of fetal hypoxia incorporating recurrence plot with convolutional neural network," *Frontiers Physiol.*, vol. 10, p. 255, Mar. 2019.

[56] A. Arratia and E. Sepúlveda, "Convolutional neural networks, image recognition and financial time series forecasting," in *Proc. Workshop Mining Data Financial Appl.* Cham, Switzerland: Springer, 2020, pp. 60–69.

[57] J.-F. Chen, W.-L. Chen, C.-P. Huang, S.-H. Huang, and A.-P. Chen, "Financial time-series data analysis using deep convolutional neural networks," in *Proc. 7th Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 87–92.

[58] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 759–766.

[59] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[60] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[61] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Open-set domain adaptation in machinery fault diagnostics (don't short) using instance-level weighted adversarial learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7445–7455, Nov. 2021.

[62] J. Tian, D. Han, H. R. Karimi, Y. Zhang, and P. Shi, "Deep learning-based open set multi-source domain adaptation with complementary transferability metric for mechanical fault diagnosis," *Neural Netw.*, vol. 162, pp. 69–82, May 2023.

[63] X. Yu, Z. Zhao, X. Zhang, X. Chen, and J. Cai, "Statistical identification guided open-set domain adaptation in fault diagnosis," *Rel. Eng. Syst. Saf.*, vol. 232, Apr. 2023, Art. no. 109047.

[64] H. Wang, Z. Xu, X. Tong, and L. Song, "Cross-domain open set fault diagnosis based on weighted domain adaptation with double classifiers," *Sensors*, vol. 23, no. 4, p. 2137, Feb. 2023.

[65] P. Panareda Busto, A. Iqbal, and J. Gall, "Open set domain adaptation for image and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 413–429, Feb. 2020.

[66] S. Zhang, Q. Zhang, J. Gu, L. Su, K. Li, and M. Pecht, "Visual inspection of steel surface defects based on domain adaptation and adaptive convolutional neural network," *Mech. Syst. Signal Process.*, vol. 153, May 2021, Art. no. 107541.

[67] A. Fuentes, S. Yoon, T. Kim, and D. S. Park, "Open set self and across domain adaptation for tomato disease recognition with deep learning techniques," *Frontiers Plant Sci.*, vol. 12, Dec. 2021, Art. no. 758027.

[68] F. I. Eyiokur, D. Yaman, and H. K. Ekenel, "Domain adaptation for ear recognition using deep convolutional neural networks," *IET Biometrics*, vol. 7, no. 3, pp. 199–206, May 2018.

[69] J. Zhou, B. Jing, Z. Wang, H. Xin, and H. Tong, "SODA: Detecting COVID-19 in chest X-rays with semi-supervised open set domain adaptation," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 5, pp. 2605–2612, Sep. 2022.

[70] M. Abdelwahab and C. Busso, "Supervised domain adaptation for emotion recognition from speech," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5058–5062.

[71] J. Zhang, J. Liu, L. Shi, B. Pan, and X. Xu, "An open set domain adaptation network based on adversarial learning for remote sensing image scene classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Sep. 2020, pp. 1365–1368.

[72] S. Nirmal, V. Sowmya, and K. P. Soman, "Open set domain adaptation for hyperspectral image classification using generative adversarial network," in *Inventive Communication and Computational Technologies*, G. Ranganathan, J. Chen, and Á. Rocha, Eds. Singapore: Springer, 2020, pp. 819–827.

[73] G. A. Richards, G. J. Morris, D. W. Shaw, S. A. Keeley, and M. J. Welter, "Thermal pulse combustion," *Combustion Sci. Technol.*, vol. 94, nos. 1–6, pp. 57–85, Nov. 1993.

[74] R. V. Bethi S. V. Gali, "Response analysis of a pitch–plunge airfoil with structural and aerodynamic nonlinearities subjected to randomly fluctuating flows," *J. Fluids Struct.*, vol. 92, Jan. 2020, Art. no. 102820.

[75] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Phys. Rep.*, vol. 438, nos. 5–6, pp. 237–329, Jan. 2007.

[76] N. Marwan, "How to avoid potential pitfalls in recurrence plot based data analysis," *Int. J. Bifurcation Chaos*, vol. 21, no. 4, pp. 1003–1017, Apr. 2011.

[77] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A, Gen. Phys.*, vol. 33, no. 2, pp. 1134–1140, Feb. 1986.

[78] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A, Gen. Phys.*, vol. 45, no. 6, p. 3403, Mar. 1992.

[79] L. Sadouk, "CNN approaches for time series classification," in *Time Series Analysis-Data, Methods, and Applications*. London, U.K.: IntechOpen, 2018, pp. 1–23.

[80] S. Chaudhary, S. Taran, V. Bajaj, and A. Sengur, "Convolutional neural network based approach towards motor imagery tasks EEG signals classification," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4494–4500, Jun. 2019.

[81] K. Jaiswal and D. Kalpeshbhai Patel, "Sound classification using convolutional neural networks," in *Proc. IEEE Int. Conf. Cloud Comput. Emerg. Markets (CCEM)*, Nov. 2018, pp. 81–84.

[82] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

S. AKSHAY received the B.Tech. degree in aerospace engineering and the M.Tech. degree in computational engineering and networking from Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India, in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in data-driven modeling in complex systems with the Center for Computational Engineering and Networking (CEN). His research interests include machine learning, deep learning, and data-driven modeling of dynamical systems.

E. A. GOPALAKRISHNAN received the Ph.D. degree from the Indian Institute of Technology Madras, Chennai, India, in 2016. He is currently the Chairperson of the Amrita School of Computing, Amrita Vishwa Vidyapeetham, Bengaluru, India. His research interests include complex systems, data driven modeling and analysis, artificial intelligence, early warning signals for catastrophic transitions, and time series analysis.

V. SOWMYA received the master's degree in remote sensing and wireless sensor networks from the Center for Computational Engineering and Networking (CEN), Amrita School of Engineering, Coimbatore, in 2011, and the Ph.D. degree in artificial intelligence (AI) for natural scene analysis from Amrita Vishwa Vidyapeetham. Her research interests include AI for signal and image analysis, biomedical, agriculture, and ecology.

J. VENKATRAMANI received the Ph.D. degree from the Department of Applied Mechanics, IIT Madras, India. He is currently with the Department of Mechanical Engineering, Shiv Nadar IoE, as an Assistant Professor. His research interests include encompasses aeroelasticity, nonlinear dynamics, and random vibrations.

DHEERAJ TRIPATHI received the M.Tech. degree in mechanical engineering from the National Institute of Technology Calicut, Kerala, India. He is currently pursuing the Ph.D. degree with the Department of Mechanical Engineering, Shiv Nadar Institution of Eminence, Greater Noida, India. His research interests include nonlinear time series analysis, nonlinear FSI, stochastic modeling, synchronization theory, and damage analysis.

JAY SHANKAR PRASAD received the B.Tech. degree in mechanical engineering from Techno India, Salt Lake, West Bengal, India, in 2017, and the M.Tech. degree in mechanical engineering from the National Institute of Technology Durgapur, West Bengal, where he is currently pursuing the Ph.D. degree. His research interests include pulse combustion, heat transfer, solar energy, fluid mechanics, and turbulent flows.

SIRSHENDU MONDAL received the Ph.D. degree from Jadavpur University, India, in 2015. He was a Postdoctoral Researcher with IIT Madras, India, and then joined Amrita Vishwa Vidyapeetham, Amritapuri Campus, India, as an Assistant Professor. He is currently an Assistant Professor with the Mechanical Engineering Department, National Institute of Technology Durgapur, India. His research interests include combustion instability, heat transfer, nonlinear dynamics, and time series analysis.